Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Espacios de reducción en sistemas de reescritura no-secuenciales e infinitarios

Tesis presentada para optar por el título de Doctor de la Universidad de Buenos Aires, en el área **Ciencias de la Computación**

Carlos Alberto Lombardi

Directores de tesis: Alejandro Norberto Ríos
Delia Nora Kesner
Eduardo Augusto Bonelli

Consejero de estudios: Alejandro Norberto Ríos

Buenos Aires, 2014

Fecha de defensa: 7 de noviembre de 2014

Reduction spaces in non-sequential and infinitary rewriting systems

We study different aspects related to the reduction spaces of diverse rewriting systems. These systems include features which make the study of their reduction spaces a far from trivial task. The main contributions of this thesis are: (1) we define a multistep reduction strategy for the *Pure Pattern Calculus*, a non-sequential higher-order term rewriting system, and we prove that the defined strategy is normalising; (2) we propose a formalisation of the concept of standard reduction for the *Linear Substitution Calculus*, a calculus of explicit substitutions whose reductions are considered modulo an equivalence relation defined on the set of terms, and we obtain a result of *uniqueness* of standard reductions for this formalisation; and finally, (3) we characterise the equivalence of reductions for the infinitary, first-order, left-linear term rewriting systems, and we use this characterisation to develop an alternative proof of the *compression* result.

We remark that we use *generic models of rewriting systems*: a version of the notion of *Abstract Rewriting Systems* is used for the study of the Pure Pattern Calculus and the Linear Substitution Calculus, while a model based on the concept of *proof terms* is used for the study of infinitary rewriting. We include extensions of both used generic models; these extensions can be considered as additional contributions of this thesis.

Keywords:
Rewriting
Standardisation
Normalising reduction strategies
Equivalence of reductions
Pattern calculi
Calculi with explicit substitutions
Infinitary rewriting
Abstract Rewriting Systems
Proof terms

Espacios de reducción en sistemas de reescritura no-secuenciales e infinitarios

En esta tesis estudiamos distintos aspectos ligados al espacio de reducción de diversos sistemas de reescritura. Los sistemas abarcados presentan características que hacen que el estudio de sus espacios de reducción diste de ser una tarea sencilla.

Las principales contribuciones son: (1) se define una estrategia de reducción multipaso para el *Pure Pattern Calculus*, un cálculo con patrones no-secuencial, y se demuestra que dicha estrategia es normalizante; (2) se propone un criterio para formalizar el concepto de reducción standard en el *Linear Substitution Calculus*, un cálculo de sustituciones explícitas cuyas reducciones se consideran módulo una relación de equivalencia sobre su conjunto de términos, obteniéndose un resultado de *unicidad* de reducciones standard para el criterio definido; y (3) se caracteriza la equivalencia entre reducciones para los sistemas de reescritura de términos *infinitarios* de primer orden y lineales a izquierda, utilizándose esta caracterización para desarrollar una demostración alternativa del resultado de *compresión*.

Destacamos el uso de *modelos genéricos de sistemas de reescritura*: se utiliza una formulación de *Sistemas Abstractos de Reescritura* para estudiar el *Pure Pattern Calculus* y el *Linear Substitution Calculus*, y un modelo basado en *proof terms* para estudiar la reescritura infinitaria. Esta tesis incluye asimismo extensiones de los dos modelos genéricos utilizados, que pueden considerarse contribuciones adicionales de la misma.

Palabras clave:
Reescritura
Estandarización
Estrategias de reducción normalizantes
Equivalencia entre reducciones
Cálculos con patrones
Cálculos con sustituciones explícitas
Reescritura infinitaria
Sistemas abstractos de reescritura
Proof terms

Espaces de réductions dans les systèmes de réécriture non-séquentiels et les systèmes de réécriture infinitaires

On aborde dans cette thèse certaines propriétés formelles de systèmes de réécriture relatives à leurs espaces des dérivations. Les calculs choisis présentent des caractéristiques particulières qui font l'étude des propriétés choisies des défis intéressants. Les contributions les plus importantes de ce travail sont: (1) nous définissons une stratégie de réduction multiradicaux pour le *Pure Pattern Calculus*, un calcul d'ordre supérieur non-séquentiel, et nous prouvons que cette stratégie est normalisante; (2) nous proposons une manière de formaliser le concept de réduction standard pour le *Linear Substitution Calculus*, un calcul avec substitutions explicites agissant à distance, dont les réductions sont considérés modulo une relation d'équivalence dans l'ensemble des termes, et nous aboutissons à des résultats d'existence et d'unicité des réductions standards pour cette formalisation; (3) nous donnons une caractérisation de l'équivalence entre les réductions pour les systèmes de réécriture des termes infinitaires du premier ordre linéaires à gauche, et nous nous servons de cette caractérisation pour développer une preuve d'une version renforcée du résultat de compression des réductions infinitaires.

Un aspect commun à ces trois sujets est l'utilisation de *formalismes génériques de systèmes* de réécriture. L'étude sur le Pure Pattern Calculus et celui concernant le Linear Substitution Calculus reposent sur le concept de *Système Abstrait de Réécriture*. D'autre part, pour le travail sur la réécriture infinitaire, on se sert d'un modèle fondé sur la notion de *proof term*. Des extensions à ces formalismes génériques sont des contributions additionnelles de cette thèse.

Mots-clés:
Réécriture
Standardisation
Stratégies de réduction normalisantes
Équivalence entre réductions
Calculs avec motifs
Calculs avec substitutions explicites
Réécriture infinitaire
Systèmes Abstraits de Réécriture
Proof terms

Agradecimientos

Dentro de la enorme cantidad de personas que contribuyeron, en formas más o menos indirectas, a que haya llegado a este punto, no puedo dejar de mencionar a **Roel de Vrijer**. Sus clases, sus comentarios, nuestras charlas, durante los meses en que estuvo en Buenos Aires en 2006, fueron un impulso importante para que me terminara de decidir a orientarme a la investigación. Después tuve la suerte de compartir con él muchos otros momentos, allá en varias visitas mías a la VU Amsterdam, y acá en visitas suyas a la UBA. El material que se incluye en esta tesis, tanto desde lo concreto (con él y con Alejandro Ríos desarrollamos el material sobre reescritura infinitaria que forma el capítulo 5, a partir de una sugerencia original de Roel, y tomando varias de sus ideas e intuiciones), como desde la forma de trabajo, le debe mucho a esos encuentros. Dank U wel Roel.

Tal vez existan tesis sin directores; aseguro que este no es el caso.

Tuve la gran fortuna de haber encontrado directores muy presentes, que estuvieron ahí para sostenerme, intercambiar pareceres, ayudarme en cuestiones del trabajo de tesis, y también en otras cuestiones, ajenas al doctorado pero importantes para mí. También para trabajar codo a codo, el "ahí" en el que los percibí está lejos de la vanidad de la "autoridad académica", lo cual les agradezco muy en particular, a los tres.

Lo que pueda tener esta tesis de profundo, de preciso, de claro, de riguroso, se debe en gran parte a sus intervenciones.

En principio, decidí embarcarme en este doctorado por una mezcla de ganas de lanzarme a la experiencia de trabajar en investigación, a ver cómo me resultaba, y las exigencias de la carrera docente. Entendía nebulosamente que "un doctorado lleva mucho tiempo", un tiempo que se mide ¡en años!, donde durante todo este tiempo se interactúa con un director. Al darme cuenta de esto, tomé mi primer decisión: si voy a trabajar tanto tiempo con una persona, que sea **Alejandro Ríos**.

Había trabajado con él durante la tesis de doctorado que compartimos con Enrique Vetere y que él dirigió. Nos encontramos, Enrique y yo, con una persona con la que fue un placer compartir ratos de trabajo, que propuso un trato cordial y cercano desde el principio, que nos acompañó muchísimo, nos fue llevando en la comprensión del tema que nos propuso, y nos guió muy bien cuando pensamos en un momento haber fracasado.

Ahora creo que nos conocemos mucho más con Ale. De esta experiencia, más amplia, más profunda, más transformadora que la anterior, una de mis suertes es haber podido disfrutar de estas características suyas, mucho más en extenso y en detalle.

Ale, gracias por haber contribuido a que estos años me hayan resultado agradables. Por la paciencia para ayudarme a desentrañar ideas en momentos en que estaban muy difusas en mi cabeza, en que sólo las podía balbucear, más de una vez pudimos transformar entre los dos tales marañas en hilos coherentes. Por la rigurosidad que aportás y que exigís en todo lo que pase por tus manos. Por la capacidad de poder hablar con vos sobre cuestiones en las que tenemos visiones parecidas, y también sobre otras que nos encuentran en posturas muy distantes. Y entre muchas otras cosas …

… por haberme presentado a **Delia Kesner**.

Trabajando con ella tuve la oportunidad de apreciar su visión abarcadora de los distintos aspectos del "trabajo de científico", en particular sobre cómo trazar y cómo seguir una lína de trabajo. También de ver el aspecto de las personas que trabajan

(¡trabajamos!) en esto; la ciencia es, finalmente, lo que hacen los científicos. Sus comentarios y visiones contribuyeron mucho con mi formación, máxime teniendo en cuenta que yo venía de otro ámbito, habiendo trabajado en desarrollo de software durante 15 años antes de volcarme a la investigación en teoría.

También aproveché su lucidez para encontrar problemas y sugerir caminos para resolverlos, o para seguir pensando, en el terreno confuso y resbaladizo que configuran ciertas demostraciones y definiciones complejas con las que trabajamos; así como su talento para organizar información, plantear la forma inicial de un artículo y después encarar la redacción de los pasajes más difíciles.

Gracias Delia por todo eso. Y muy especialmente, por haberme recibido mucho mejor que bien en mis períodos en París, ayudándome en todo, desde las llamadas telefónicas en francés hasta prestarme un colchón, y ocupándose de que los trámites necesarios avanzaran. En particular, por la magia que hiciste que posibilitó que pasara un período allá cuando lo necesité por razones personales.

En esta temática y en Buenos Aires, es inevitable interactuar con **Eduardo Bonelli**. De Edu aprecié muchísimo la capacidad de conectar distintos trabajos, de armar marcos conceptuales en los que se inscribe el problema que se está estudiando, de recordar (¡prodigiosamente!) la cita del artículo donde se habla de algo parecido a lo que tenemos entre manos. También la intuición (supongo que conexa con lo anterior) para proponer soluciones para los atolladeros con que nos encontramos. En particular, su sugerencia (dicha con su deliciosa sencillez, como quien sugiere echarle un poco de orégano a la salsa) de "probar con gripping" salvó lo que hoy es el capítulo 3.

Edu, gracias por el aporte. Y desde lo personal, por lo que aprendí sobre mantener el gusto por el trabajo en teoría durante las tormentas interiores provocadas por los otros millones de cosas de las que nos ocupamos en estos años, apreciando tus ganas de "hacer cuentas", charlando con vos varias veces, muchas en la Universidad de Quilmes, donde tengo la suerte de trabajar en la oficina al lado de la tuya.

Finalmente, quiero agradecerles a los tres la confianza que tuvieron en mí, que fue en muchos momentos claramente mayor a la que yo me tenía, y que fue un punto de apoyo constante, sobre todo cuando las cuentas no salen, los esquemas no cierran. Me hace feliz pensar que esta tesis es digna de semejante (por quiénes y por cuánto) confianza.

Hay otra persona a la que el contenido de esta tesis le debe mucho. Varias iluminaciones de **Beniamino Accattoli**, y su firmísimo propósito de estudiar los cálculos de sustituciones explícitas a distancia desde distintas perspectivas, son una parte primordial del sustento sobre el que se basa el capítulo 4. También tuvo mucho que ver con varios de los resultados que allí aparecen. Ti ringrazio tantissimo Beniamino.

Yendo para atrás en el tiempo, antes de la decisión de emprender esta aventura doctoral, hubo otra más fundamental: la de orientar mis actividades al campo de la docencia y la investigación. En esa decisión tuvo mucho que ver la invitación inicial que me hizo **Nicolás Passerini** para participar en el dictado de una materia en la UTN, hace ya 12 años. A partir de la experiencia inicial de los primeros cursos que compartimos, surgieron un grupo de personas y una serie de actividades que crecieron, ambos, en proporciones y formas insospechadas, más allá de lo que la imaginación más delirante hubiera podido sugerir. Una de las consecuencias de ese movimiento es que varios decidimos "pegar el salto a la academia", y acá estoy, en 2014, terminando un doctorado.

Gracias Nico, por el impulso inicial, por haber hecho lugar al principio a ese tipo "venido de afuera". Por todas las cosas que compartimos en este tiempo, que ayudaron a definir, y/o apuntalaron, muchos (realmente muchos) de los intereses y perspectivas de la persona que soy ahora, donde este doctorado es una de ellas. Y por tus/nuestros proyectos que son parte de la argamasa que forma el futuro, el mío y (sospecho) el de unas cuantas personas más.

Gracias también a todas/os las damas y los caballeros con los que compartimos aulas, cursos, mesas de picadas y de comida en sus más variadas formas, principios, ideales, ideas, sueños, y un largo etcétera. Cualquier enumeración sería pesada (por lo larga) y al mismo tiempo injusta (por lo imperfecto de mi memoria); espero que estas palabras les lleguen a unos cuantos de sus destinatarios, y que logren perdonarme por no mencionarlos explícitamente.

Un paso más atrás está la licenciatura, Exactas, mundo nuevo, gente distinta de la que había frecuentado hasta ese momento. Recién empezando, **Elvio Nabot**, compañero fantabuloso para aprender juntos a programar con objetos y avanzar en general en la comprensión de lo que se trata programar. También para compartir actividades e ideas adentro y afuera de la facultad, asomarnos desde nuestros 20 años a una Buenos Aires en una crisis económica feroz, pero todavía con el fermento cultural impulsado por la vuelta a la democracia (años 88 a 91, para quienes esta refrencia espaciotemporal les signifique algo).

Después, una combinación entre Elvio y la más pura casualidad provocaron el comienzo del encuentro con **Enrique Vetere**. Este encuentro lleva 20 años, de excavar y excavar para entendernos cada uno a sí mismo y al otro, de tomar y generar y descartar distintas formas de encontrarnos (haber sido "compañeros de facultad" fue una de ellas), de estar.

Enri y Elvio: el aliento, la confianza en mis capacidades, la escucha, y el purísimo cariño que recibí de ustedes (y familia, Vivi, Ana, Sol, Wanda, Tomás) fueron y son fundamentales (en el sentido literal ligado con "fundamento") de cada cosa importante que emprendo, entre ellas, el camino que termina en este manuscrito. Muchas muchas gracias.

A su vez, no fue trivial que una tarde de marzo del 88 entrara a la primer clase de mi Licenciatura en Computación. En el camino hasta ahí, hay dos personas insoslayables mucho más allá de su mera cercanía. **Má** (o sea Vicenta Puglia), gracias ppor inculcarme el gusto por leer y conocer desde chico. **Olimpia**, gracias por ocuparte de tantísimo más de lo que le corresponde, o lo que pudiera esperarse de, una hermana, sobre todo en tiempos complicados. Ni el paso del tiempo, ni los (tantos) cambios, hacen ni que me olvide ni que deje de apreciar lo queNi hicieron por mí, y cómo contribuyó a que haya llegado hasta acá con los éxitos que pude disfrutar en el camino.

Termina el flashback, vuelvo al doctorado, que me lleva a París. J'ai été à Paris, plus précisement au laboratoire PPS, à plusieurs reprises pendant mon doctorat.

Au PPS, ce n'est pas possible d'oublier **Odile Ainardi**, qui m'a aidé avec plein de démarches, en trouvant toujours la manière de les faire marcher, sans jamais perdre la convivialité quand on parlait dans son bureau. Je te remercie Odile!

C'était très jolie pour moi d'avoir partagé du temps, des déjeuners dans la cantine, et même quelques bières et des dîners, avec d'autres thèsards du PPS. Particulièrement, ces

interactions ont été fondamentales dans mon apprentisage de la langue française. Merci Grégoire, Fabien, Barbara, Sévérine, Thibaut, et tant d'autres, vous avez été vraiment gentiles.

Pendant mon séjour le plus long, j'ai fréquenté les leçons de ping chez l'**Association du Patronage Sainte Mélanie**. Heureusement, j'ai trouvé un groupe de personnes très sympa, avec lequels j'ai eu la chance de parler sur les sujets les plus diverses. Merci a tou(te)s!

Al poco tiempo de empezar el doctorado, ocurrió la sorpresa de conocernos con **Eve** (o sea, Evelina Maranzana). Decidir estar juntos fue rapidísimo. Siguieron unos años muy nómades en los que a veces quedábamos cruzados de continente, desde hace un tiempito ya que estamos juntos en continuo. Gracias amor por los mimos, por tantos ratos y tantas experiencias, por armar una pareja en la que la experiencia de vida nos permite partir con algunas cosas claras, por poner mis problemas en contexto, por ampliar mi imaginario, por tu mirada, por tu sonrisa, por sacudirme, por permitir, por alentar y por generar que este doctorado haya resultado una verdadera experiencia, con todas las letras. Te amo.

A Eve, que le da sabor, profundidad y sentido a esto, y a cualquier cosa en la que me embarque.

A Enza, que literalmente, me puso en este mundo.

A Olimpia, que se encargó de mí muchísimo más de lo que corresponde a una hermana.

A Enri, porque esto es una etapa de algo que empezamos juntos, este gol es para vos.

y
A la República Argentina, en cuyo contexto estoy pudiendo presentar una tesis doctoral con la que tal vez no podría ni estar soñando en circunstancias distintas.

Contents

Chapter 1

Introduction

Rewriting is the study of stepwise, i.e. gradual and discrete, transformation of objects. If the objects being transformed are *terms*, that is, well-formed strings of symbols, then we speak of *term rewriting*.

Rewriting has a significant and continuous influence in different areas of computer science. In this sense, the λ-*calculus*, one of the most ancient rewriting systems, is arguably the most influential one.

From a purely theoretical viewpoint, the λ-calculus defines a model of computation, in equal terms with Turing machines and recursive functions. This fact yields the relevance of rewriting for the theory of computer science.

On the other hand, maybe the most far-reaching contribution of rewriting in the practice of programming is that the features and simplicity of λ-calculus led to the development of the *functional programming model*, which is enjoying an increasing influence into the global computer programming community.

Besides the existence of programming languages based mainly in the functional programming model, as e.g. Lisp (`www.lispworks.com/documentation/HyperSpec/Front/index.htm`), Erlang (`www.erlang.com`), OCaml (`ocaml.org`) and Haskell (`www.haskell.org`); we remark that some features, concepts and techniques inspired by this model, such as the inclusion of "λ-expressions" (that is, anonymous functions), and the use of *generics* in type systems, have been adopted in mainstream programming languages, including Java (`docs.oracle.com/javase/specs/`), C# (`msdn.microsoft.com/en-us/library/618ayhy6.aspx`), Python (`www.python.org`) and others. The recent (first appeared in 2003) Scala language (`www.scala-lang.org`) combine several concepts common in functional programming with the main constructs of object-oriented programming; its rapidly growing popularity contributes to foster functional-related constructs and practices inside the programming community.

We also mention that some concepts coming from the functional programming languages and style, like the emphasis on the control of the effect/mutability generated by the different parts of a program, the use of higher-order functions, and the use of continuations, permeate into other programming communities.

On another front, rewriting systems offer a formal framework for the study of different aspects of programs, such as their evaluation and their type disciplines. Related to the latter, a plethora of *typed rewriting systems* have been proposed.

In this thesis, we study formal properties of rewriting systems focused on different concerns related to functional-based programming languages, including: *pattern calculi*, which model the phenomenon of *pattern matching*; *explicit substitution calculi*, oriented to the detailed study of the implementation of languages; and *infinitary rewriting systems*, which allow to study potentially infinite computations.

In the remainder of this introduction, we revisit the main concepts of rewriting, then we describe briefly the studied rewriting systems, subsequently we introduce the two *models of reduction spaces* being the main tools used in this work, and finally we comment its main contributions.

1.1 Rewriting

The origins of rewriting, and particularly of term rewriting, predate its establishment as a definite research area. Historically, the major source for the development of term rewriting is the emergence in the 1930s of the λ-calculus, together with its twin combinatory logic. Several formal properties, currently associated to the general framework of rewriting, were originally analysed for these systems. Later, the notion of term rewriting system has been formalised, and their properties studied from a general perspective. One early example is [KB70].

1.1.1 Some basic features of rewriting

A simple example of rewriting is the simplification of an arithmetic expression in order to obtain its result. In this view, the computation of the result of the expression $(1 \times 1) \times (0 \times 0)$ can be described by either of the following stepwise transformations:

$$\begin{array}{l} (1 \times 1) \times (0 \times 0) \;\to\; 1 \times (0 \times 0) \;\to\; 1 \times 0 \;\to\; 0 \\ (1 \times 1) \times (0 \times 0) \;\to\; (1 \times 1) \times 0 \;\to\; 1 \times 0 \;\to\; 0 \end{array}$$

The initial expression is simplified by means of a **sequence** of **rewrite steps**. The use of a directed arrow (instead of e.g. some equality symbol) reflect that transformations, as modeled in the theory of rewriting, have a definite direction from **source** to **target**. A **rewriting system** specifies the objects being transformed and the allowed rewrite steps; in this example, arithmetic expressions and sound simplification steps respectively. The final expression of both exhibited sequences, namely 0, cannot be further rewritten (i.e. simplified). Such objects are known as the **normal forms** of a rewriting system.

We use $t \twoheadrightarrow u$ to denote that the object u can be obtained from t through a sequence of rewrite steps.

As in this example, most applications of rewriting allow a multiplicity of rewrite sequences from a common source. This fact leads to two of the most basic concerns of rewriting:

Termination Do all possible rewrite sequences attain a normal form after a finite number of steps, or can *infinite rewrite sequences* be built?

Uniqueness of normal forms If two rewrite sequences having the same source end in normal forms, is it possible to assert that those normal forms coincide?

Let us analyse the consequence of not enjoying either of these properties in our example about computing the result of arithmetic expressions. The lack of *uniqueness* *of normal forms* would imply a basic inconsistency: different results could be obtained from a common expression, depending on how the computation from that expression is carried on. The lack of *termination* would imply that certain computations could run indefinitely without yielding a result.

In many applications of rewriting, both termination and uniqueness of normal forms are desired properties. There are important exceptions to this observation though. We mention the examples of *CCS* [Mil99] and the π-*calculus* [SW01], which are adequate models of concurrent computations, despite the fact that they do not enjoy neither of these two properties.

Another property referred repeatedly in the literature is the **confluence** or **Church-Rosser property**. A rewriting system is *confluent* iff, whenever $t \twoheadrightarrow u_1$ and $t \twoheadrightarrow u_2$, there exists an object s verifying $u_1 \twoheadrightarrow s$ and $u_2 \twoheadrightarrow s$. That is: in a confluent rewriting system, given two sequences of rewrite steps from a common source, a common target can always be obtained by further rewriting them, thus “joining” the two original sequences. Observe that in a confluent rewriting system, if $t \twoheadrightarrow u_1$, $t \twoheadrightarrow u_2$, and both u_1 and u_2 are normal forms, then necessarily $u_1 = u_2$, because the only object verifying $u_1 \twoheadrightarrow s$ is u_1, and similarly for u_2. Hence confluence implies uniqueness of normal forms. In fact, proving confluence is a way to obtain uniqueness of normal forms for a rewriting system.

We end this brief informal description of the field of rewriting, by noticing that in many cases, the stepwise transformation of objects is specified by **rewrite rules**. These rules encode the schemas of the allowed transformations: each rewrite step must correspond to the **application of a rule**. A set of rewrite rules form the basis of a **rewriting system**. In our example about simplification of arithmetic expressions, taken from [vO94], the rules:

$$1 \times x \;\to\; x \qquad\qquad x \times 0 \;\to\; 0$$

suffice to justify each of the steps in both of the rewrite sequences exhibited. E.g. the step $(1 \times 1) \times (0 \times 0) \;\to\; 1 \times (0 \times 0)$ corresponds to an application of $1 \times x \;\to\; x$, where x stands for the second occurrence of 1 from the left. Observe that the rule application in this step does not involve all the source term, but only the **subterm** 1×1. Usually, rules can be applied to either a complete object or only to a part of it.

Notice the use of **variables** in rules. If we allow variables to stand, not just for numbers, but for arbitrary expressions, then the following rewrite sequences are also allowed:

$$(1 \times 1) \times (0 \times 0) \;\to\; 1 \times (0 \times 0) \;\to\; 0 \times 0 \;\to\; 0$$
$$(1 \times 1) \times (0 \times 0) \;\to\; (1 \times 1) \times 0 \;\to\; 0$$

1.1.2 Common rewriting terminology

In the rewriting literature, the terms **reduction step** and **reduction sequence** are commonly used for “rewrite step” and “sequence of rewrite steps” respectively. Reduction steps are sometimes called just **reductions**. The underlying view is that suggested in the given example. Rewriting is considered as the stepwise simplification, or *reduction* of an initial expression. In many situations, the goal of rewriting is to attain *normal forms*.

Another common term in rewriting is *reducible expression*, or **redex**. A redex is usually defined as any part of a term, which makes the term to be subject of a reduction step, together with the corresponding rewrite rule. Considering the rewriting system of the previous section, the term $(1 \times 1) \times 0$ includes two redexes, one for the subterm 1×1 and the rule $1 \times x \ \rightarrow \ x$, and the other for the whole term, and the rule $x \times 0 \ \rightarrow \ 0$. The **contractum** is the term resulting of the contraction of a redex, e.g. the contractum of 1×2 is 2. The expression '**redex occurrence**' is used to distinguish different parts of the same term being instances of the left-hand sides of rewrite rules, even when the applicable rule, and possibly also the instance, coincide. E.g. the term $(1 \times 3) \times (1 \times 3)$ includes two redex occurrences, both corresponding to instances of the left-hand side of the rule $1 \times x \rightarrow x$ having the form 1×3.

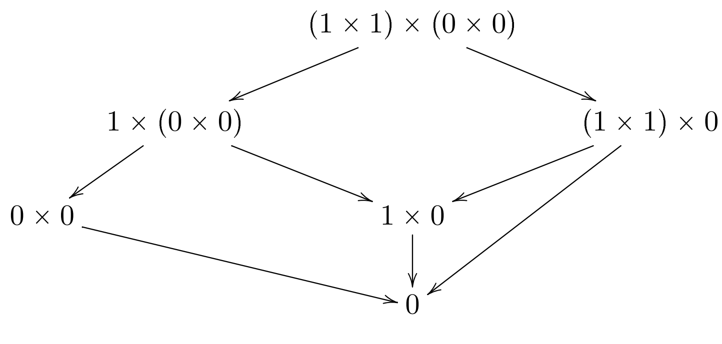
There is an obvious correspondence between the concepts of *redex*, more precisely *redex occurrence* and *reduction step*, the main difference being in their respective focus. The word "redex", and the expression "redex occurrence", denote the fact that a transformation step can be performed on a certain term, while "reduction step" stands for the act of performing that step, or put in other words, of **contracting** a redex. We use the terms "redex occurrence" and "step" interchangeably in this thesis.

Finally, we mention the notion of **set of coinitial steps**, which is simply a set of steps which share their source object.

1.1.3 Reduction spaces

The transformations modeled by a rewriting system can be described by means of a directed graph, whose nodes are the objects and whose edges are the reduction steps. This graph is mentioned as the **reduction space** (or *derivation space*) of the system in, e.g., [KG97], [HL91] and [Mel96]. The reduction sequences are exactly the paths in the reduction space. The normal forms correspond to the nodes with no outgoing edges. The pairs of connected objects form the **reduction relation** of a rewriting system: the pair $\langle t, u \rangle$ is in the relation iff $t \twoheadrightarrow u$. Hence, a reduction space is more detailed than the corresponding reduction relation.

Complex reduction spaces can correspond to even simple rewriting systems. The following figure depicts the portion of the reduction space of the rewriting system of our example, including just the sequences having $(1 \times 1) \times (0 \times 0)$ as source term.



Some of the concepts and properties usually studied in rewriting are closely related with reduction spaces. This is the case for the *equivalence between reduction sequences*, the *standardisation* properties, and the study of *reduction strategies*. We describe briefly these concepts in the following.

An analysis of the **equivalence between reduction sequences** is usually a good guide to the understanding of complex reduction spaces. Two sequences are commonly considered equivalent if they represent the same reduction activity, performed in different order. A simple example are the sequences $(1 \times 1) \times (0 \times 0) \to 1 \times (0 \times 0) \to 1 \times 0$ and $(1 \times 1) \times (0 \times 0) \to (1 \times 1) \times 0 \to 1 \times 0$: they include the same steps, performed in the two possible orders. This situation corresponds exactly to the upper diamond in the previous figure.

It is not true in general that any two reductions sharing their source and target are equivalent. A simple example can be given in the rewriting system about simplification introduced earlier. Consider the term $1 \times (1 \times 1)$. This term includes two redexes, both for the rule $1 \times x \to x$; contracting either of them yields 1×1. The situation is illustrated in the following figure, where each occurrence of 1 in the source term $1 \times (1 \times 1)$ is given a different *label*, and for each step, the corresponding subterm in its source term is indicated with a brace, and the replacement for x is indicated by underlining

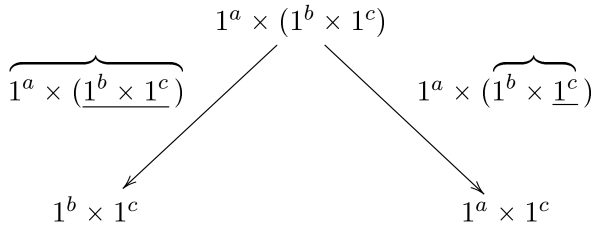


Figure 1.1: Two confusing steps

The resulting reduction sequences, both consisting in just one step, are *not* equivalent.

The aim of the study of **standardisation** is to find subsets of the set of reduction sequences of a rewriting system covering all the reduction relation. Namely, an adequate characterisation of a class of **standard reduction sequences**, shorthand `s.r.s.`, should enjoy the following condition: whenever $t \twoheadrightarrow u$, there is a `s.r.s.` having t and u as source and target respectively. In terms of the reduction space, a class of `s.r.s.` is a set of paths covering all the pairs of connected objects.

For any rewriting system, an obvious class of `s.r.s.`, namely the one including *all* the reduction sequences in the system, exists. The interesting classes of `s.r.s.` are those as narrow as possible, the best being those enjoying a *uniqueness* condition: whenever $t \twoheadrightarrow u$, there is *exactly one* `s.r.s.` having t and u as source and target respectively. If equivalence of reductions is considered, then the uniqueness condition can be rephrased as the existence of exactly one `s.r.s.` for each class of equivalent reductions.

In the literature, e.g. [CF58, Klo80, GLM92, Mel96, BKdV03], standardisation is related with the notion of **external** step: in a `s.r.s.`, external steps should precede internal ones. E.g., in the term $1 \times (2 \times 0)$, the step $\underbrace{1 \times (2 \times 0)} \to 2 \times 0$ should precede $1 \times (\underbrace{2 \times 0}) \to 1 \times 0$. Therefore, the reduction sequence $1 \times (2 \times 0) \to 2 \times 0 \to 0$ is standard, while $1 \times (2 \times 0) \to 1 \times 0 \to 0$ is not.

A **reduction strategy** can be described as a "plan" indicating how reduction should proceed from a given term. A strategy can be defined as a function: for any object t not in *normal form*, it indicates a reduction step, or in some cases a set of reduction steps, having t as source. A target object is obtained by following the indication given by the strategy, i.e., by performing the selected step(s). In turn, applying the strategy to this target object yields a step/a set of steps to be further performed; and so on.[1]

We name as **multistep** reduction strategies, those indicating more than one step for at least one object. In such cases, some way of performing the selected steps *simultaneously* must be defined.[2] The notion of **complete development** of a set of coinitial steps is usually involved with the task of simultaneous contraction.

The aim when defining a reduction strategy is to arrive at normal forms, whenever it is possible, by its *systematic application*; that is, by following the "plan" given by the strategy. More formally, a strategy $\mathcal{S}$ is **normalising** if, whenever $t \twoheadrightarrow u$ and u is a normal form, there is a reduction sequence $t = t_0 \twoheadrightarrow t_1 \twoheadrightarrow \ldots \twoheadrightarrow t_{n-1} \twoheadrightarrow t_n = u$, where for all i, the reduction $t_i \twoheadrightarrow t_{i+1}$ is the result of applying the indication given by $\mathcal{S}$ for t_i.[3] We use the term **normalisation** to refer to the study of how normal forms can be computed, involving the definition of normalising reduction strategies, and also the techniques to prove that a reduction strategy is normalising.

Standardisation and normalisation are among the subjects of this thesis.

1.1.4 The λ-calculus and higher-order term rewriting systems

As noted in the beginning of this introduction, the λ-calculus [Chu32, CR36, Chu41, Bar84] is arguably the most influential rewriting system. This calculus was developed prior to, and greatly influenced, the emergence of the general study of rewriting. Several of the main concepts, techniques and results studied in rewriting appeared previously applied to the particular case of the λ-calculus.

The λ-calculus can be described as a minimalist formalisation of the mechanism by which a function is applied to an argument. Its syntax includes just the elements needed to describe function application: variables, the abstraction constructor to define functions, and application to link a function definition with an argument. Numbers and arithmetical operands will be used in the following as well, to favor a more intuitively appealing description.

Let us consider the term

$$(\lambda x.x + x + x)3$$

denoting the application of the function $(\lambda x.x + x + x)$ to the argument 3. The occurrences of the variable x in the subterm $x + x + x$ are **bound** by the abstraction λx. Term rewriting systems including, like the λ-calculus, some mechanism to bind variable occurrences, are known as **higher-order term rewrite systems**. Conversely, in **first-order term rewrite systems**, no such mechanisms are present.

> **A note about terminology**: in the literature about rewriting, the name "term rewriting system", and specially the acronym "TRS", refer usually to

[1] In the more general case of *non-deterministic* reduction sequences, more than one indication can be given for the same object, so that any of those indications may be followed. In this thesis, only *deterministic* reduction sequences will be considered.

[2] We remark the difference between a multistep reduction strategy and a non-deterministic strategy. In the former, a set of steps form *a single indication*, so that all these steps are supposed to be performed simultaneously. In the latter, there can be several different indications, and *only the step(s) in one of them* are supposed to be performed.

[3] In this characterisation of normalising reduction strategies, *uniqueness of normal forms* is assumed. In the general case, it suffices to obtain *one of the normal forms* which can be reached from t, by following the strategy.

first-order systems. We will use the name "term rewriting system" to refer to the set of *all* systems, either first- or higher-order, and the explicit form "first-order term rewriting system" when needed.

Several general formats for higher-order term rewriting system have been proposed, we mention *CRS* [Klo80], *HRS* [Nip91, MN98] and *ERS* [GKK05].

The λ-calculus includes just one rewrite rule, the β-rule, namely

$$(\lambda x.s)u \quad \longrightarrow \quad \{x := u\}\, s$$

where $\{x := u\}\, s$ stands for the **substitution**, in the term s, of the (non-bound) occurrences of x with the term u. An example of a rewrite step in the λ-calculus follows:

$$(\lambda x.x + x + x)\, 3 \quad \longrightarrow \quad 3 + 3 + 3$$

Observe that this is an *atomic* step in the model given by λ-calculus: the application of the substitution $\{x := 3\}$ in $x + x + x$ is considered as an external operation. *Explicit substitution* calculi, cfr. Section 1.2.2, arise as a way to model the substitution operation within a rewriting system, providing specific rules to describe how a substitution is applied to a term.

Results about standardisation and normalising reduction strategies for the λ-calculus are present in the literature since [CF58]. Afterwards, other works including standardisation studies for λ-calculus have appeared, we mention [Bar84] Sec 11.4, [Tak95], [Kas00] and [Cra09]. The notions of *call-by-name*, *call-by-value* and *call-by-need* reduction strategies, cfr. [Plo75], characterise different families of reduction strategies for the λ-calculus; these notions frame, in many cases, the way in which the evaluation of a program should proceed, a relevant aspect in the design of a programming language.

Finally, we list some acronyms for λ-calculus terms to be used in this manuscript: I for $\lambda x.x$, K for $\lambda x.(\lambda y.x)$, D for $(\lambda x.xx)$, and Ω for DD. Observe that $\Omega \to \Omega$, this being the only step having Ω as source term.

1.2 The rewriting systems studied in this thesis

1.2.1 Pattern calculi

Let us revisit the rule of the λ-calculus:

$$(\lambda x.s)u \quad \longrightarrow \quad \{x := u\}\, s$$

We notice that the rule applies to *any* abstraction and *any* argument: there is no way to restrict, or *filter*, the set of arguments that are accepted by a given abstraction. Moreover, the abstractions have a *unique body*: the λ-calculus does not include mechanisms to define functions allowing different bodies for different kinds of arguments.

This situation does not coincide with the common practice of functional programming languages. These languages include **pattern matching** features, allowing to specify restrictions to the possible arguments of a function, and also to give different definitions of the same function to arguments having different features. A simple example is the following definition of the length of a list in Haskell:

```
length []      = 0
length (x:xs)  = 1 + length xs
```

The function `length` requires its argument to be a list, and moreover it has two defining clauses, for empty (denoted `[]`) and non-empty (denoted `x:xs`) lists respectively.

These observations lead to the development of **pattern calculi**; rewriting systems aiming to provide explicit formalisations of different forms of pattern matching. Several of these calculi provide some sort of "generalised abstraction", say having the form

$$\lambda p.s$$

where p is a **pattern**. This is the case of the λ-calculus with patterns [vO90, KvOdV08], the ρ-calculus [CK98, CK01], the pattern calculus [Jay04, Jay09] and the Pure Pattern Calculus [JK06a, JK09].[4] The set of the valid patterns includes all the variables, implying that the defined calculus can be considered as a generalisation of the λ-calculus.

The rewrite rule is generalised accordingly to

$$(\lambda p.s)u \longrightarrow \{p/u\}\, s$$

where $\{p/u\}$ is the result of **matching** the argument u against the pattern p. E.g., if we represent pairs as a data structure whose constructor, a constant $\mathtt{p}$, is applied successively to the left and right components of a pair, as in $\mathtt{p}\ 3\,4$, then the following should be a valid rewrite step for a pattern calculus:

$$(\lambda \mathtt{p}\, x\, y.\, y)\ (\mathtt{p}\ 3\,4) \ \rightarrow\ 4$$

If there is no possible matching, as in

$$(\lambda \mathtt{p}\, x\, y.\, y)\ 3$$

then the resulting term is not a redex, unless the calculus provides some **error mechanism** to deal with such cases.

The two main issues when devising a pattern calculus are the definition of the set of valid patterns, and subsequently, that of the argument/pattern matching. A *too liberal* choice of the set of patterns (as e.g. accepting any term as a valid pattern), combined with a naïve definition of matching, would break confluence, and thus uniqueness of normal forms; cfr. [vO90, KvOdV08, CF07, JK09]. On the other hand, a *too restrictive* choice of the pattern set would hinder the possibility of modeling interesting phenomena related with pattern matching. This observation led to the definition of several different pattern calculi.

The **Pure Pattern Calculus**, whose shorthand is `PPC`, is one of the rewriting systems studied in this thesis. In `PPC`, any term can be a pattern. Particularly, a pattern can include free variable occurrences, and reduction steps can occur inside patterns as well. These features allow *dynamic pattern building.*

The `PPC` is described in Section 3.4.1. We give here some examples of its features, using a simplified version of its syntax. To allow the pattern of an abstraction to include

[4] In other proposals, as in the λ-calculus with constructors [AMR06, AMR09], the filter and the possibility of multiple clauses are modeled by a *case* construct. The Basic Pattern Matching Calculus, [Kah03, Kah04] combines both generalised abstraction and case.

free variable occurrences, a set of bounded variables is attached to the abstractor λ, written below the λ. E.g., the identity function can be defined in PPC by the term $\lambda_{\{x\}}x.x$. Consider the following valid term in PPC:[5]

$$t = (\lambda_{\{x\}}x.(\lambda_{\{y,z\}}x(yz).y))$$

In this term, both occurrences of x are bound by the outer abstractor, the one including x in its set of bound variables.

By giving an appropriate argument to t, we produce a concrete function out of the generic function specification $(\lambda_{\{y,z\}}x(yz).y)$. E.g., if $\mathtt{a}$ is a constructor, then the following reduction sequence

$$(\lambda_{\{x\}}x.(\lambda_{\{y,z\}}x(yz).y))\,\mathtt{a}\,(\mathtt{a}(34)) \to (\lambda_{\{y,z\}}\mathtt{a}(yz).y)\,(\mathtt{a}(34)) \to 3$$

shows that the application of t to a constructor produces a function which accepts, as arguments, only data structures on that constructor. Moreover, if the argument given to t is in turn a function, that function is applied to the argument (yz) *inside the pattern* of $\lambda_{\{y,z\}}x(yz).y$. Therefore, we obtain a further flexibility for the construction of the pattern. Check the following reduction sequence:

$$\begin{aligned}(\lambda_{\{x\}}x.(\lambda_{\{y,z\}}x(yz).y))(\lambda_{\{x',y'\}}x'y'.\mathrm{p}y'x')(\mathrm{p}34)&\\ \to (\lambda_{\{y,z\}}(\lambda_{\{x',y'\}}x'y'.\mathrm{p}y'x')(yz).y)\,(\mathrm{p}34)&\\ \to (\lambda_{\{y,z\}}\mathrm{p}zy.y)\,(\mathrm{p}34) \to 4&\end{aligned}$$

where the second reduction step is performed inside a pattern, as suggested previously. As a consequence of these features, forms of *polymorphism* not present in programming languages currently used in software development, can be expressed in PPC, cfr. [JK09] where several examples are given. The just described examples show that patterns in PPC can be **dynamic**.

We remark that a carefully defined matching operation allows PPC to handle patterns like $x'y'$, as shown in the previous examples. E.g., given the matching rules of PPC, in the term

$$(\lambda_{\{x,y\}}xy.x)(\underbrace{(\lambda_{\{z\}}z.z)3})$$

the only redex is the one indicated by the brace: a pattern like xy does *neither match nor fail* w.r.t. an argument being a redex, thus preventing the loss of confluence, and consequently of uniqueness of normal forms.

1.2.2 A finer step granularity – Explicit Substitution calculi

To motivate the introduction of *explicit substitution calculi*, let us revisit this λ-calculus rewrite step:

$$(\lambda x.x + x + x)\,3 \;\to\; 3 + 3 + 3$$

In the model of the rewrite space given by the λ-calculus, the transformation of $(\lambda x.x + x + x)\,3$ to $3 + 3 + 3$ is considered as a single, atomic rewrite step. On the other hand, this transformation can be regarded as a complex operation, involving the *replacement*

[5] In fact, in the simplified variant used in this introduction. The actual PPC term corresponding to this example involves the use of *matchables*, as described in Section 3.4.1

of each occurrence of x in the body $x + x + x$ with the argument 3, and (depending of the desired detail level) also the *search* for those occurrences inside the body.

The view of substitution as a complex operation is particularly appropriate for the study of the *implementation of functional programming languages*. Indeed, it is not surprising that substitution is deeply involved in the evaluation of a functional program, since functional programming has its roots in the λ-calculus. As a consequence, the implementation of functional programming languages are faced with the task of computing substitutions, a task revealed to be far from trivial in practice. Hence the need of formal models reflecting explicitly the complexity of the substitution operation.

This situation motivated the emergence of variations of the λ-calculus widely known as **explicit substitution calculi**, shorthand **ES calculi**. We describe the main features of these rewriting systems, using the $\lambda\mathtt{x}$ calculus, [Ros92, BR95] to illustrate them.

The syntax of the ES calculi includes a construct to explicitly denote substitutions. If s and u are terms, then

$$s\,[x/u]$$

is a valid term as well in the $\lambda\mathtt{x}$ calculus.

A rule analogous to that of the λ-calculus is present. The expression subject to rewrite is the same: the application of an abstraction to an argument. But in this case, the rule only *generates* the corresponding substitution, without executing it:

$$(\lambda x.s)u \;\longrightarrow\; s\,[x/u]$$

Additional rewrite rules model how a substitution is performed. For the $\lambda\mathtt{x}$ calculus, these rules are:[6]

$$\begin{array}{rcll} (t_1 t_2)\,[x/u] & \longrightarrow & (t_1\,[x/u])(t_2\,[x/u]) & \\ (\lambda y.t)\,[x/u] & \longrightarrow & \lambda y.t\,[x/u] & \\ x\,[x/u] & \longrightarrow & u & \\ y\,[x/u] & \longrightarrow & y & \text{if } y \neq x \\ \mathtt{c}\,[x/u] & \longrightarrow & \mathtt{c} & \end{array}$$

The first and second rule allow to *propagate* an explicit substitution through a term, generating copies in the process. As a result, each copy is either applied or erased, by virtue of the third or fourth rule respectively.

Assuming two constants $\mathtt{p}$ and $\mathtt{s}$, the λ-calculus reduction step $(\lambda x.\mathtt{p}x(\mathtt{s}x))\,3 \;\rightarrow\; \mathtt{p}\,3(\mathtt{s}\,3)$ can be simulated in $\lambda\mathtt{x}$ as follows:

$$\begin{array}{llll} (\lambda x.\mathtt{p}x(\mathtt{s}x))\,3 & & & \\ \rightarrow & (\mathtt{p}x(\mathtt{s}x))\,[x/3] & \rightarrow & (\mathtt{p}x)\,[x/3]\,((\mathtt{s}x)\,[x/3]) \\ \rightarrow & \mathtt{p}\,[x/3]\,x\,[x/3]\,((\mathtt{s}x)\,[x/3]) & \rightarrow & \mathtt{p}\,[x/3]\,x\,[x/3]\,(\mathtt{s}\,[x/3]\,x\,[x/3]) \\ \rightarrow & \mathtt{p}\,x\,[x/3]\,(\mathtt{s}\,[x/3]\,x\,[x/3]) & \rightarrow & \mathtt{p}\,3\,(\mathtt{s}\,[x/3]\,x\,[x/3]) \\ \rightarrow & \mathtt{p}\,3\,(\mathtt{s}\,x\,[x/3]) & \rightarrow & \mathtt{p}\,3\,(\mathtt{s}\,3) \end{array}$$

Figure 1.2: Simulation of a λ-calculus step in the $\lambda\mathtt{x}$ calculus

[6] In fact, the syntax of $\lambda\mathtt{x}$ does not include constants; they are added to give the examples shown in this section.

Observe that this is just one of the many possible reduction sequences in $\lambda\mathtt{x}$ simulating the given λ-calculus step.

The possibility of having several explicit substitutions in the same term can further complicate the reduction space of an ES calculus. Consider the following examples:

$$\begin{array}{ll}
(\lambda y.(\lambda x.\mathtt{p}x(\mathtt{s}y))\,y)\,3 & (\lambda y.(\lambda x.\mathtt{p}x(\mathtt{s}y))\,y)\,3 \\
\quad \to \;\; (\,(\lambda x.\mathtt{p}x(\mathtt{s}y))\,y)\,[y/3] & \quad \to \;\; (\,(\lambda x.\mathtt{p}x(\mathtt{s}y))\,y)\,[y/3] \\
\quad \to \;\; (\,(\mathtt{p}x(\mathtt{s}y))\,[x/y])\,[y/3] & \quad \to \;\; (\,(\lambda x.\mathtt{p}x(\mathtt{s}y))\,[y/3])\,(y\,[y/3]) \\
 & \quad \to \;\; (\lambda x.\,(\mathtt{p}x(\mathtt{s}y))\,[y/3])\,(y\,[y/3]) \\
 & \quad \to \;\; ((\mathtt{p}x(\mathtt{s}y))\,[y/3])\,[x/y\,[y/3]]
\end{array}$$

Figure 1.3: Two reduction sequences from the same source in the $\lambda\mathtt{x}$ calculus

The reduction spaces of ES calculi turns out to be extremely complex, leading to difficulties to obtain a calculus simultaneously satisfying a series of properties related with confluence, termination, and *simulation* of the λ-calculus (namely, the ability of simulate in an ES calculus any reduction sequence $t \twoheadrightarrow u$ in the λ-calculus). This situation implied the development of many different ES calculi, including [HL89, ACCL91, KR95, BBLRD96, DG01, Kes07].

More recently, a different approach to ES calculi has been proposed. The **ES calculi at a distance**, [Mil07a, AK10, Acc12] are based on the idea of avoiding the propagation of explicit substitutions through a term, allowing a substitution to be applied to a distant variable occurrence. These calculi include a rule of the shape:

$$C[\![x]\!]\,[x/u] \;\longrightarrow\; C[\![u]\!]\,[x/u] \tag{1.1}$$

where $C[\![\,]\!]$ is an arbitrary *context* including a (free) occurrence of x. In these calculi, explicit substitutions do not move: replacements are performed without any need to propagate them. This fact leads, in principle, to simpler reduction spaces.

The **linear substitution calculus**, [ABKL14], an ES calculus at a distance, is one of the rewriting systems studied in this thesis. We will use the shorthand $\lambda_{\mathtt{lsub}}^{\sim}$ to refer to this calculus. It is both a slight generalisation of a calculus by Robin Milner [Mil07a], related to bigraphs, from which it inherits the *substitution rule* at a distance (1.1), and a slight modification of the structural λ-calculus presented in [AK10], related to *proof-nets*. This calculus adds a rule to erase "useless" explicit substitutions, corresponding to the idea of *garbage collection*:

$$t\,[x/u] \;\longrightarrow\; t \qquad\qquad \text{if } x \notin \mathtt{fv}(t)$$

The λ-calculus reduction step $(\lambda x.\mathtt{p}x(\mathtt{s}x))\,3 \;\to\; \mathtt{p}\,3(\mathtt{s}\,3)$ can be simulated in $\lambda_{\mathtt{lsub}}^{\sim}$ as follows:[7]

$$\begin{array}{rlll}
(\lambda x.\mathtt{p}x(\mathtt{s}x))\,3 & \to\;\; (\mathtt{p}x(\mathtt{s}x))\,[x/3] & \to & (\mathtt{p}\,3\,(\mathtt{s}x))\,[x/3] \\
 & \to\;\; (\mathtt{p}\,3\,(\mathtt{s}\,3))\,[x/3] & \to & (\mathtt{p}\,3\,(\mathtt{s}\,3))
\end{array}$$

Figure 1.4: Simulation of a λ-calculus step in $\lambda_{\mathtt{lsub}}^{\sim}$

[7] As indicated above for the $\lambda\mathtt{x}$, we added constants to the syntax of $\lambda_{\mathtt{lsub}}^{\sim}$, described in Section 4.1, for the examples shown in this section.

This simulation is indeed simpler that the one shown for the $\lambda\mathtt{x}$ calculus, cfr. Figure 1.2.2. Moreover, the multiplication of different reduction sequences simulating the same λ-calculus step is more limited than in $\lambda\mathtt{x}$.

On the other hand, to enhance the analogy between $\lambda_{\mathtt{lsub}}^{\sim}$ and proof-nets, the definition of the calculus includes three equivalence equations, which model the fact that substitution constructs must be considered as somewhat "floating" in a term, their actual positions in a term being irrelevant to a certain extent. E.g., the following equation

$$t[x/s][y/u] \approx t[y/u][x/s] \qquad \text{if } x \notin \mathtt{fv}(u) \text{ and } y \notin \mathtt{fv}(s)$$

models the idea that (in principle) the order of substitutions in a substitution chain is irrelevant. Two terms related by the equivalence relation generated by these equations, can be considered as different descriptions of an unique object being rewritten. This fact poses a challenge for the study of the calculus.

1.2.3 Infinitary rewriting

Let us consider the rewriting systems T_1 and T_2 defined as follows. The system T_1 includes the number 1, the addition symbol, a unary functor symbol l, the list constructor denoted by the colon, and the rule

$$l(x) \longrightarrow x : l(x+1)$$

The system T_2 includes two constants a and b, and the rules

$$a \longrightarrow b \qquad\qquad b \longrightarrow a$$

Both T_1 and T_2 are *non-terminating* rewriting systems, since both allow infinite rewrite sequences. For T_1 we have

$$l(1) \to 1 : l(2) \to 1 : 2 : l(3) \to 1 : 2 : 3 : l(4) \to \ldots$$

where $2, 3, \ldots$ are shorthand for $1+1$, $1+1+1$, etc.. On the other hand, the following is a rewrite sequence for T_2:

$$a \to b \to a \to b \to a \to \ldots$$

Even though both sequences can run indefinitely long without yielding a final result, a relevant difference can be observed. Consider the *sequences of partial results*, which are respectively

$$\langle 1 : l(2), 1 : 2 : l(3), 1 : 2 : 3 : l(4), \ldots \rangle \qquad \text{and} \qquad \langle b, a, b, a, \ldots \rangle$$

It is not difficult to grasp that while the former sequence *converges* to the *infinite* list of natural numbers, namely $1 : 2 : 3 : 4 : \ldots$, the latter is a *divergent* sequence.

Interestingly, the rules of both systems can be easily rendered in a functional programming language. Using Haskell, we can define

```
natlist n = n : natlist (n+1)
diva      = divb
divb      = diva
```

The evaluation of these functions behaves as suggested by analysing the corresponding rewriting systems. While the evaluation of `natlist 1` generates the list `[1,2,3,4...` , the evaluation of `diva` runs indefinitely without producing any partial result.[8]

These considerations motivate the study of **infinitary (term) rewriting systems**; cfr. [KKSdV90, DKP91, KKSdV95], and [BKdV03] Ch. 12. In these systems, both the terms being rewritten and the rewrite sequences can be infinite. Convergence is a central concept in the study of infinitary rewriting: the study of properties of the reduction space is mostly focused on convergent reductions.

Different convergence criteria have been proposed. In this thesis, **strong convergence**, as defined in [KKSdV95], is used. For a reduction sequence to be strongly convergent, it does not suffice to obtain ever-growing fixed prefixes, but it is also required that the sequence formed by the *depth* (i.e. distance to the root) of each step in the sequence tends to infinity. If we consider the rule $f(x) \to f(g(x))$, the sequence

$$f(a) \to f(g(a)) \to f(g(g(a))) \to \dots$$

is *not* strongly convergent, because all its steps are head steps.

The requirement about depths, added in the strong convergence criterion, is crucial for the characterisation of equivalence of infinitary reductions we present in Section 5.3, which is consequently valid for strongly convergent reductions only. Furthermore, other interesting properties, as the compression result (see below), and the possibility of defining projections, also hold only for strongly convergent reduction sequences, as mentioned in [KKSdV95, BKdV03, KdV05], where strong convergence is favored. These considerations lead to the adoption, in this thesis, of the strong convergence criterion.

If an infinite rewrite sequence converges, via its sequence of partial results, to a certain term, we consider that term as the (infinitary) *target* of that sequence. Following some of the existent literature, we write

$$t \twoheadrightarrow\!\!\!\!\!\rightarrow u$$

to denote that the term u is the target of a convergent rewrite sequence having t as source. Then the example for the system T_1 can be described as follows

$$l(1) \twoheadrightarrow\!\!\!\!\!\rightarrow 1 : 2 : 3 : 4 : \dots$$

The concept of *termination* can be extended to infinitary rewriting as follows: a rewrite sequence is **infinitarily terminating** iff it either yields a final result (as in finitary rewriting), or it converges to an *infinitary result.* The infinite rewrite sequence given for the system T_1 is infinitarily terminating. The concepts of *confluence* and *uniqueness of normal forms* can be extended to infinitary rewriting analogously. Several results, both positive and negative, of the extension of well-known properties of finitary rewriting into its infinitary counterpart are present in the literature, we mention [Ken92, KKSdV95, KdV05, Zan08, EGH$^+$10, EHK12].

Another well-known result about infinitary rewriting is **compression**, cfr. [KKSdV90, KKSdV95, BKdV03, Ket12]. To motivate it, let us add the pair construct, denoted by angle brackets, into T_1, and consider the following rewrite sequence

$$\langle l(1), l(1)\rangle \to \langle 1 : l(2)\, ,\, l(1)\rangle \to \langle 1 : 2 : l(3)\, ,\, l(1)\rangle \to \langle 1 : 2 : 3 : l(4)\, ,\, l(1)\rangle \to \dots$$

[8] The `div` in the names `diva` and `divb` are for "divergent".

This sequence converges to $\langle 1:2:3:4\ldots,\ l(1)\rangle$. On the other hand, this result is not final, the reduction sequence can continue as follows:

$$\langle 1:2:3:4\ldots,\ l(1)\rangle \to \langle 1:2:3:4\ldots, 1:l(2)\,\rangle \to \langle 1:2:3:4\ldots, 1:2:l(3)\,\rangle \to \ldots$$

The infinitary final result $\langle 1:2:3:4\ldots, 1:2:3:4\ldots\rangle$ can be obtained by resorting *again* to convergence.

This situation is modeled in infinitary rewriting, by considering reduction sequences whose length go beyond the first infinite ordinal, ω. In the example, the two reduction sequences shown can be concatenated obtaining a rewrite sequence having length $\omega \times 2$. We obtain

$$\langle l(1), l(1)\rangle \twoheadrightarrow \langle 1:2:3:4\ldots,\ l(1)\rangle \twoheadrightarrow \langle 1:2:3:4\ldots, 1:2:3:4\ldots\rangle$$

The compression property states that the restriction of rewrite sequences to the first infinite ordinal does not affect the power of infinitary rewriting. Formally, for any t, u terms, if $t \twoheadrightarrow u$, then there is a reduction sequence whose length is at most ω having t and u as source and target respectively. The following reduction sequence, having length ω, coincides in source and target with that having length $\omega \times 2$ just shown:

$$\begin{aligned}\langle l(1), l(1)\rangle &\to \langle 1:l(2)\,,\, l(1)\rangle \to \langle 1:l(2)\,,\, 1:l(2)\rangle \\ &\to \langle 1:2:l(3)\,,\, 1:l(2)\rangle \to \langle 1:2:l(3)\,,\, 1:2:l(3)\rangle \\ &\to \langle 1:2:3:l(4)\,,\, 1:2:l(3)\rangle \to \langle 1:2:3:l(4)\,,\, 1:2:3:l(4)\rangle \\ &\twoheadrightarrow \langle 1:2:3:4\ldots, 1:2:3:4\ldots\rangle\end{aligned}$$

1.3 Generic models of rewriting systems

The features of the different rewriting systems introduced in Sec. 1.2 show the great diversity of term rewriting systems present in the literature.

In spite of this diversity, there are some basic notions common to all of them: term, reduction step, redex, reduction sequence, reduction space. There are also some properties whose study is interesting for many term rewriting systems, as equivalence of reductions, standardisation, or normalising reduction strategies.

These similarities motivate the definition and study of *generic models* of rewriting systems. A generic model allows for abstract definitions of notions, and for abstract proofs of properties, about rewriting systems. The defined notions and the proved properties are, thus, valid for any rewriting systems which fits into the model.

Two generic models, of different nature, are used in this thesis. They are described in the following.

1.3.1 Abstract Rewriting Systems

Several abstract models of transformation, which apply to rewriting systems, have been proposed in the literature; we mention those presented in [New42, Hin69], [Bar84] Chapter 3, [BKdV03] Chapter 1, and [BKdV03] Chapter 8.2. We use in this thesis a model, first presented in [GLM92], and later refined by Paul-André Melliès in [Mel96, Mel05]. In this proposal, a rewriting system is modeled as a structure named **Abstract Rewriting System**, shorthand **ARS**; we refer to this abstract model of rewriting as the **ARS**

model. In the following, we introduce the version described in [Mel96], the one used in this thesis.

The definition of an ARS is based on two sets, that of the *objects* being rewritten, notation $\mathcal{O}$, and that of the rewriting *steps*, notation $\mathcal{R}$.[9] In this model, no detail is included about the structure, or any other intrinsic aspect, of objects. Each step is modeled primarily as the link between a *source* object and a *target* object, defined by means of two functions $\mathsf{src}, \mathsf{tgt} : \mathcal{R} \to \mathcal{O}$.

In this way, any term rewriting system can be modeled as an ARS, by considering terms and rewriting steps as the sets of objects and steps respectively.

The letters a, a', a_1, b, c, etc. will be used to denote steps, and we will sometimes decorate the arrow denoting a step by the name given to that step, e.g. $\xrightarrow{b}$.

We notice that the identification of steps allows to describe adequately situations like that shown in Fig. 1.1, page 5. Modeling that case by an ARS would yield two different steps, sharing their source and target objects: $\overbrace{1 \times (\underbrace{1 \times 1}_{b})}^{a} \xrightarrow{a} \underbrace{1 \times 1}_{b}$. and $\overbrace{1 \times (\underbrace{1 \times 1}_{b})}^{a} \xrightarrow{b} \underbrace{1 \times 1}_{a}$.

Additional information is modeled through a number of *relations* defined on steps.

The **residual relation** is a ternary relation; the notation $a[\![b]\!]a'$ denotes that the triple (a, b, a') is in the residual relation. Residuals are related with the *tracing* of steps. A triple $a[\![b]\!]a'$ indicates that the step a' is a direct correlate, in the target of b, of the step a present in the source of b. We say in this case that a' is a residual of a after b.

As an initial example, let us consider the step $(1 \times 1) \times (0 \times 0) \xrightarrow{b} 1 \times (0 \times 0)$ in the rewriting system about arithmetic expressions. It is intuitively clear that the step corresponding to 0×0 in the term $1 \times (0 \times 0)$ is a direct correlate of the step corresponding to the same subterm in $(1 \times 1) \times (0 \times 0)$. If we name these redexes as a' and a respectively, then we have $a[\![b]\!]a'$. The following figure depicts this situation

$$(\underbrace{1 \times 1}_{b}) \times (\underbrace{0 \times 0}_{a}) \xrightarrow{b} 1 \times (\underbrace{0 \times 0}_{a'})$$

Figure 1.5: A simple example of residuals

In Fig. 1.5, we identify a step a with its *redex*, i.e. the corresponding subterm in its source term; cfr. Section 1.1.2. This convention is used subsequently throughout this thesis. In Fig. 1.6, we show different cases in which the behavior of steps w.r.t. residuals is less straightforward. The three examples included in this figure verify $a[\![b]\!]a'$, in Fig. 1.6:(b) we have $a[\![b]\!]a''$ as well.

[9] Called *redexes* (in French "radicaux") in [Mel96], hence the reason why we use the letter $\mathcal{R}$.

a) $\overbrace{(\lambda x.3)(\underbrace{(\lambda y.y)5}_{a})}^{b} \xrightarrow{b} 3$ b) $\overbrace{(\lambda x.xx)(\underbrace{(\lambda y.y)5}_{a})}^{b} \xrightarrow{b} (\underbrace{(\lambda y.y)5}_{a'})\ (\underbrace{(\lambda y.y)5}_{a''})$

c) $\overbrace{(\lambda x.\,\underbrace{(\lambda y.y)x}_{a})5}^{b} \xrightarrow{b} \underbrace{(\lambda y.y)5}_{a'}$

Figure 1.6: Examples of residuals in the λ-calculus

A step a can have no, or several, residuals after another step b, as shown in Fig. 1.6:(a) and (b). In turn, Fig. 1.6:(c) shows that the subterm corresponding to a step can differ from that of a residual: the subterm of the step a, $(\lambda y.y)x$, is "transformed" into $(\lambda y.y)5$ by the contraction of b.

The set of objects, that of steps together with the source and target functions, and the residual relation, form a minimal version of the definition of an ARS.

Noticeably, *equivalence of reduction sequences* can be studied in the obtained model. In Fig. 1.7, we revisit the example of equivalence given previously, now decorated by giving names to the participating terms and steps.

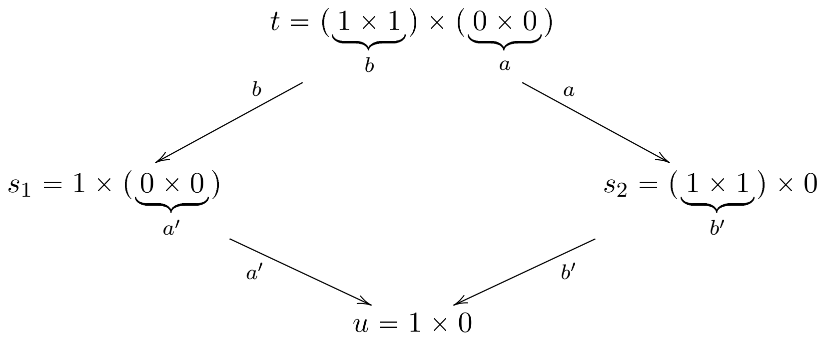


Figure 1.7: Equivalence of reductions

In the figure, a' is the only residual of a after b, and analogously, b' is the only residual of b after a. The steps a and b do not interfere with each other in this example: the effect of performing the residual of a after b (on the term s_1) can be considered as equivalent to that performing the original step a (on t).[10] This observation indicates that the steps a and b are **orthogonal**[11], and therefore, that the shown reduction sequences are *equivalent*: they consist of a followed by the residual of b, and b followed by the residual of a, respectively.

Observe that in the example of Fig. 1.7, permuting in either reduction depicted the

[10]The equivalence can be further verified by giving *labels* to suitable symbols in the terms, as done when introducing equivalence of reduction sequences.

[11]We remark that this characterisation of orthogonality, based on the behavior of steps and residuals, differs from that resulting of a more *syntactic* approach, based on the form of rewrite rules. Notably, orthogonality of rewrite steps, and in general of rewriting systems, can be studied without making reference to set of rules of the latter.

order in which the steps a and b are performed, yields the other one.[12] In the ARS model, the notion of **step permutation** leads to the formal definition of the equivalence of reductions: two reduction sequences are considered *equivalent* if either of them can be obtained from the other by means of a sequence of step permutations.

We remark that the definition of an ARS *involves only the identity* of objects and steps. No syntactic information is included in this model. Considering the example shown in Fig. 1.7, an ARS modeling the arithmetic simplification rewriting system would include four objects and four steps, which can be given the names t, s_1, s_2, u and a, b, a', b' for this description, which satisfy the following: $\mathsf{src}(a) = \mathsf{src}(b) = t$, $\mathsf{tgt}(b) = \mathsf{src}(a') = s_1$, $\mathsf{tgt}(a) = \mathsf{src}(b') = s_2$, $\mathsf{tgt}(a') = \mathsf{tgt}(b') = u$, $a[\![b]\!]a'$, and $b[\![a]\!]b'$.

In spite of the expressive features of the residual relation, the binary **embedding** relation on steps, notation $<$, must be considered as well for most interesting uses of the ARS model. Embedding provides a *partial order* between steps having the same source.

The intent of the pair $b < a$ is to denote that b has some direct power over a, which is reflected in the residuals of a after b. A possible form of this power would be that b can **erase** or **duplicate** a, i.e. to make a have no, or several, residuals after b. As the λ-calculus examples shown in Fig. 1.6 suggest, this power is, in many cases, related with the fact that b actually *nests* a, namely, that the subterm corresponding to b encompasses that of a, as in Fig. 1.6. Indeed, when modeling the λ-calculus as an ARS, a possible definition of the embedding coincides exactly with nesting as it was just defined.[13]

The concepts of step, residual and embedding yield a model focused on the study of the *reduction space* of the modeled rewriting system. Cfr. [Mel96], pg. 70:

> The abstract approach allows to study a (rewriting) system through the derivation space it induces.[14]

ARS equipped with the residual and embedding relations are rich enough to develop fully **abstract proofs**, notably about standardisation (cfr. Section 2.1.8) and normalising reduction strategies. In an abstract proof, only the information pertaining to the ARS model is used in order to prove some statement. In turn, the statement subject of an abstract proof usually correspond to the following pattern:

> any ARS, provided that it verifies some **axioms**, enjoys a certain property.

Some of the axioms describe basic properties of the residual relation, while others describe comparisons between the embedding of some steps and that of their corresponding residuals. The axioms can be said to provide an abstract characterisation of the residual and embedding relations.

An example of axiom regarding residuals follows:

[12]If a has more than one residual after b, or vice versa, then the permutation of a and b is not as simple as shown in Fig. 1.7. Cfr. Section 2.1.7 for details.

[13]Observe that, in fact, the step $(\lambda x.s)u \to \{x := u\}s$ can only erase or duplicate steps lying inside u. This observation leads to a second possible model of the λ-calculus as an ARS, considering a restricted embedding relation. The properties of (the ARS yielding from) both "full-nesting" and restricted embeddings are studied in [Mel96].

[14]In the French original: "L'approche abstraite permet de traiter un système à partir de l'espace des dérivations qu'il induit."

Ancestor Uniqueness $\quad b_1[\![a]\!]b'$ and $b_2[\![a]\!]b' \Rightarrow b_1 = b_2$.

This axiom expresses the condition that the step b' cannot be residual of two different steps at once, after the contraction of a.

The following axiom express a condition involving residuals and embedding:

Context freeness $\quad b[\![a]\!]b' \wedge c[\![a]\!]c' \Longrightarrow a < c \vee (b < c \Leftrightarrow b' < c')$

This axiom indicates a necessary condition, namely $a < c$, to allow a to break the invariance in the embedding relation between two redexes b and c, w.r.t. their respective residuals, that is, to "dissolve" the embedding between b and c in their residuals, or to "create", between the residuals of b and c, an embedding which did not exist before.

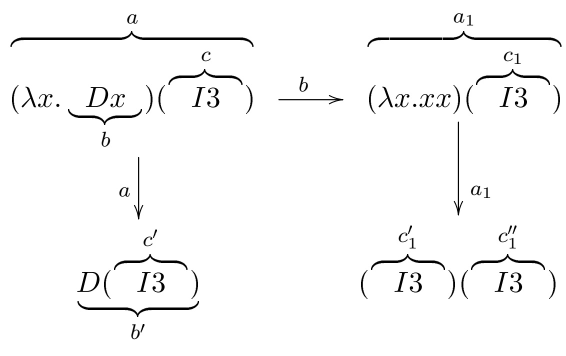
Regarding the use of ARS in this work, modeling the *linear substitution calculus* as an ARS equipped with the residual and embedding relations suffices to obtain the standardisation results we aim at.

On the other hand, the work on *Pure Pattern Calculus* requires an extended version of the ARS model, involving a third relation on steps. We introduce the **gripping** relation by means of an example in λ-calculus. Let us consider the following step, where we tag other steps and their residuals

$$\overbrace{(\lambda x.\underbrace{Dx}_{b})(\overbrace{I3}^{c})}^{a} \xrightarrow{a} \underbrace{D(\overbrace{I3}^{c'})}_{b'}$$

In the situation depicted, $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Observe that $b' < c'$, while neither of their origins, b and c respectively, embed the other one. Besides a embedding both b and c in the original term $(\lambda x.Dx)(I3)$, there is another factor crucial for this change in relative embeddings: the subexpression corresponding to b, Dx, includes an occurrence of the variable x, bound by the abstraction $\lambda x.Dx$. The replacement of this occurrence of x by $I3$ provokes the appearance of a new embedding on the residuals.

There is another consequence, particularly harmful for the work on the *Pure Pattern Calculus*, of this relation between b and a. Observe the following diagram



Observe that c has *one* residual after a. This situation changes for the respective residuals after b: now there are *two* residuals of c_1 after a_1. This change in the number of residuals affects, in a critical way, a measure used in one of the main proofs of this work.

Thus the need to consider the gripping relation. In the example, we say that the step b *grips* the step a. We will avoid the use of *gripping* steps, in the situations where invariance of a measure related to numbers of residuals is required. We give some details when describing the results of this work, at the end of this Section.

1.3.2 Proof terms

The concept of *proof term* provides another generic model of reduction spaces. It is a model less abstract than that given by ARS: the structure of the objects being rewritten is involved, and the rules play a fundamental role. On the other hand, it keeps more information about the modeled reduction sequences.

Several versions of the proof term model have been developed for λ-calculus in [Hil96], for first-order term rewriting systems in [BKdV03], and for a generic formalism of higher-order term rewriting systems in [Bru08]. The brief description which follows is based in the first-order term rewriting version.

A **proof term** is the representation of a reduction sequence as a term, using an enlarged set of symbols. Indeed, the language of the proof terms for a given term rewriting system includes all the symbols in that system, plus the **rule symbols**, which indicate the application of rules in a reduction sequence. There is one rule symbol for each rule, its arguments corresponding to the variables occurring in the rule.

The two rules given for the arithmetic simplification rewriting system would therefore correspond to two rule symbols, let us name them μ and ν. The relation between each rule symbol and its corresponding rule can be described as follows

$$\mu(x) : 1 \times x \;\to\; x \qquad\qquad \nu(x) : x \times 0 \;\to\; 0$$

We introduce the notion of proof term by means of some examples. We show three proof terms denoting single rewriting steps in the arithmetic simplification rewriting system, along with the step corresponding to each one. In each case, the subterm being affected by the step is indicated with an upper brace, and the subterm corresponding to the argument of the rule symbol is underlined.

$$\begin{array}{rcl} \mu(3) & : & \overbrace{1 \times \underline{3}} \;\to\; 3 \\ \mu(1) \times (0 \times 0) & : & (\overbrace{1 \times \underline{1}}) \times (0 \times 0) \;\to\; 1 \times (0 \times 0) \\ 1 \times \nu(1 \times 1) & : & 1 \times (\overbrace{(\underline{1 \times 1}) \times 0}) \;\to\; 1 \times 0 \end{array}$$

In order to denote reduction sequences, the binary (infix) symbol $\cdot$, denoting **concatenation**, or **composition**, of steps, is added. A proof term of the form $A \cdot B$ denotes the reduction sequence represented by A, followed by that represented by B. We show some proof terms along with the reduction sequences they denote. Concatenation being associative, we omit brackets in the last example.

$$\begin{array}{rcl} 3 \times \nu(2 \times 1) \;\cdot\; \nu(3) & : & 3 \times ((2 \times 1) \times 0) \;\to\; 3 \times 0 \;\to\; 0 \\ \mu(1) \times (0 \times 0) \;\cdot\; 1 \times \nu(0) & : & (1 \times 1) \times (0 \times 0) \;\to\; 1 \times (0 \times 0) \;\to\; 1 \times 0 \\ \mu(1) \times (0 \times 0) \;\cdot\; 1 \times \nu(0) \;\cdot\; \mu(0) & : & (1 \times 1) \times (0 \times 0) \;\to\; 1 \times (0 \times 0) \;\to\; 1 \times 0 \;\to\; 0 \end{array}$$

Moreover, rule symbols and concatenation can be combined in different ways, allowing to denote specifically that several steps are performed *simultaneously*, and/or that some sequence of steps is *localised* in some part of a term. Some examples follow:

$$\begin{array}{rcl}
\mu(1) \times \nu(0) & : & (1 \times 1) \times (0 \times 0) \mathrel{-\!\!\!\circ\!\!\!\rightarrow} 1 \times 0 \\
\mu(\nu(2)) & : & 1 \times (2 \times 0) \mathrel{-\!\!\!\circ\!\!\!\rightarrow} 0 \\
\mu(2 \times (\nu(3)) \cdot \nu(2) & : & 1 \times (2 \times (3 \times 0)) \mathrel{-\!\!\!\circ\!\!\!\rightarrow} 2 \times 0 \to 0 \\
2 \times \big(\mu(1) \times 3 \cdot \mu(3)\big) & : & 2 \times ((1 \times 1) \times 3) \to 2 \times (1 \times 3) \to 2 \times 3
\end{array}$$

where the symbol $\mathrel{-\!\!\!\circ\!\!\!\rightarrow}$ denotes that a number of steps are simultaneously performed.

Observe that the proof term $\mu(1) \times \nu(0)$, which describe the *simultaneous* application of the two steps in the term $(1 \times 1) \times (0 \times 0)$, is different from either $\mu(1) \times (0 \times 0) \cdot \nu(0)$ or $(1 \times 1) \times \nu(0) \cdot \mu(1) \times 0$, which describe the reduction sequences comprising the same steps in any of the two possible orderings. In general, proof terms denoting simultaneous or localised reduction, as the ones just described, are different from those denoting the reduction sequences comprising exactly the same steps, in any possible sequential order. This fact indicates that the proof term model allows to distinguish subtle differences in the way in which rewriting steps are applied.

In this document, we will use the name **contraction activity** to denote the whole set of possibilities in which reduction steps can be combined. As we have just seen, the proof term model allows to describe many different forms of contraction activity, including but not limited to reduction sequences.

Equivalence of reductions is defined in this model by means of equational reasoning on proof terms: two reduction sequences are equivalent if the proof terms denoting them can be proven equivalent in the congruence generated by some equation schemas.

By means of the obtained equational logic, cfr. Section 2.2.3,[15] (a proof term denoting) the concatenation of two orthogonal steps can be "packed", obtaining (another proof term denoting) their *simultaneous* contraction. Reciprocally, a simultaneous contraction can be "unpacked", obtaining the sequential concatenation of its component steps. The permutation of two adjacent steps can be modeled by "packing" them, and subsequently "unpacking" them in reverse order.

Let us illustrate this idea by means of an example. The reduction sequences

$$\begin{array}{l}
(\underbrace{1 \times 3}) \times (2 \times 0) \;\to\; 3 \times (\underbrace{2 \times 0}) \;\to\; 3 \times 0 \\
(1 \times 3) \times (\underbrace{2 \times 0}) \;\to\; (\underbrace{1 \times 3}) \times 0 \;\to\; 3 \times 0
\end{array}$$

are equivalent: the same two steps are contracted, in a different order. These reduction sequences are denoted by the proof terms

$$(\mu(3) \times (2 \times 0)) \cdot (3 \times \nu(2)) \qquad \text{and} \qquad ((1 \times 3) \times \nu(2)) \cdot (\mu(3) \times 0)$$

respectively. The equivalence of these proof terms is obtained by means of the following abridged (i.e. not all the details are included) judgement:

$$(\mu(3) \times (2 \times 0)) \cdot (3 \times \nu(2)) \;\approx\; \mu(3) \times \nu(2) \;\approx\; ((1 \times 3) \times \nu(2)) \cdot (\mu(3) \times 0)$$

Notice that the concatenation of the two steps is "packed", obtaining $\mu(3) \times \nu(2)$, and subsequently this simultaneous contraction is "unpacked" in the other order, yielding

[15]The characterisation of the equivalence of reductions through equational reasoning on proof terms is extended in this thesis to infinitary rewriting, cfr. Section 5.3.

$(1 \times 3) \times \nu(2) \ \cdot \ \mu(3) \times 0$. Observe that we can establish, not only the equivalence of the two reduction sequences, but also that both reduction sequences are equivalent to the simultaneous contraction of its two steps.

On the other hand, the equivalence of the following reduction sequences:

$$1 \times (\underbrace{2 \times 0}) \to 1 \times 0 \to 0 \qquad\qquad \underbrace{1 \times (2 \times 0)} \to 2 \times 0 \to 0$$

which involve two *nested* steps, can be verified analogously, as follows:

$$1 \times \nu(2) \ \cdot \ \mu(0) \quad \approx \quad \mu(\nu(2)) \quad \approx \quad \mu(2 \times 0) \ \cdot \ \nu(2)$$

We end this brief description of the proof term model by indicating that in [BKdV03] a second characterisation of equivalence for the proof term model, based on the concept of the *residuals* of one proof term after another, is described. A third characterisation, based on *tracing*, uses proof terms as one of its ingredients. The three characterisations are proven equivalent for first-order term rewriting systems.

1.4 Outline of the contributions

Three directions of work were pursued in this thesis, regarding respectively the *Pure Pattern Calculus*, the *linear substitution calculus*, and the class of *first-order infinitary rewriting systems*.

In all cases, the rewriting systems are analysed by means of a generic model: the ARS model for the Pure Pattern Calculus and the linear substitution calculus, the proof term model for infinitary rewriting systems. Also in all cases, the work includes adaptations to the model and/or the development of new abstract proofs, needed to obtain the desired results. These adaptations and proofs are also contributions of this thesis in their own merit.

Hence this thesis can be regarded as a work about the use of generic models, to study rewriting systems whose features make the analysis of their reduction spaces a challenging task.

We describe the contributions obtained in each of the three directions.

1.4.1 Normalising reduction strategies for non-sequential calculi

The first aim of the work in this direction is to obtain a normalising reduction strategy for the *Pure Pattern Calculus*, or `PPC`. The challenge lies in `PPC` being a *non-sequential* rewriting system: there exist terms, not being normal forms, and not having any *needed* redex.[16]

A redex in a term t is said *needed* if its contraction cannot be avoided when computing a normal form for t. That is, if for any reduction $t \twoheadrightarrow u$ where u is a normal form, either the redex, or at least one of its residuals, is included in the reduction. Several results about normalisation of reduction strategies present in the literature are based on systematic contraction of needed redexes, assuming that each term not being a normal form includes at least one needed redex. This is the case of the *leftmost-outermost*

[16] For a brief discussion about the notion of (non-)sequential rewriting systems, cfr. the introduction to Chapter 3.

reduction strategy for the λ-calculus, first studied in [CF58], and also of the theory of neededness developed in [HL91] for first-order term rewriting.

The study of the literature about non-sequential systems suggests to consider *multistep* reduction strategies; cfr. [SR93, vR97, vO99] and the study of *external* strategies in [Mel96]. On the other hand, we aimed at being not too liberal in the sets of redexes selected.

The first contribution of the work in this direction is the **definition of a multistep reduction strategy for `PPC`**. This strategy selects a single redex in many situations. Particularly, it coincides with leftmost-outermost if `PPC` is restricted to the λ-calculus.

Of course, a proof stating that the defined strategy is normalising must be developed. We favored a proof having an *abstract* flavor, ideally described in some generic model of rewriting systems. By pursuing this approach, we aim to obtain a proof which could be applied to other strategies and systems. Moreover, in the author's opinion, the development of a proof relying in abstract properties contributes to a deeper understanding of the notions participating in that proof.

An additional contribution of the work in this direction is an **abstract normalisation proof, described in the ARS model**. As indicated in Section 1.3.1, the residual, embedding and gripping relations are considered. The proof, cfr. Section 3.3, states that systematic contraction of *necessary* and *non-gripping* sets of redexes is normalising, for ARS verifying a number of axioms.

This proof was originally developed only for `PPC` ([BKLR12]). The proof we present in this thesis is the result of translating the structure of that proof to the abstract setting given by the ARS model.

The notion of necessary set of steps is a generalisation of that of needed step. A set of steps in a term t is necessary if for any reduction $t \twoheadrightarrow u$ where u is a normal form, at least one step in the set, or one of its residuals, is contracted. Systematic contraction of necessary sets of steps is proved normalising for first-order term rewriting systems in [SR93]. Our abstract normalisation proof takes the main ideas of that proof, and re-elaborates them in the broader setting given by the ARS model, in which higher-order systems can be described as well. The *non-gripping* condition is, as the name suggests, defined in terms of the gripping relation, and it is the culprit for the inclusion of that relation in the present thesis.

The abstract normalisation proof relies in all the basic, embedding and gripping axioms pertaining to the ARS model, described in Sections 2.1.3 to 2.1.6, except for one of the embedding axioms. The axiom not considered in the proof, *Stability*, describes a property related with residuals and embedding, which does not hold for non-sequential rewriting systems. Therefore excluding this axiom allows to use the ARS model to reason about such systems.

On the other hand, the abstract normalisation proof also relies in a novel axiom, described in Section 3.1.4. This axiom allows to complete the analysis of the preservation of embedding in residuals, by targeting a case complementary with those covered by the axioms included in the ARS model presentation.

As a side effect, the proof shows that some notions about (sets of) redexes and their relation with reduction sequences, can be adequately defined and handled in the ARS model.

1.4.2 Standardisation for the linear substitution calculus

Some years after the appearance of the first ES calculi, standardisation results for one of them, namely $\lambda\sigma$ [ACCL91], were presented in [Mel96]. Interestingly, the ARS model is used to study $\lambda\sigma$. Indeed, this system is one of the main examples given as applications of the ARS model in [Mel96]. In spite of the many ES calculi proposed afterwards, the author is aware of no other standardisation results for any of them.

We present in this thesis results of **existence and uniqueness of standard reductions for the linear substitution calculus**, $\lambda_{\mathtt{lsub}}^{\sim}$. We use the ARS model to study this system. In the author's opinion, the existence of these results suggests that the reduction spaces of ES calculi at a distance are indeed more manageable than those of previous ES calculi.

The ARS modeling $\lambda_{\mathtt{lsub}}^{\sim}$ verifies all the axioms required for the *standardisation existence result* in [Mel96]. Therefore, this result applies immediately to $\lambda_{\mathtt{lsub}}^{\sim}$. On the other hand, two of the axioms required for the *standardisation uniqueness result* given in [Mel96] do not hold for $\lambda_{\mathtt{lsub}}^{\sim}$. We overcome this difficulty by developing a novel proof of the standardisation uniqueness result, which does apply to $\lambda_{\mathtt{lsub}}^{\sim}$ despite the fact that it does not satisfy some of the conditions required by the statement in [Mel96]. This **abstract standardisation proof**, described in Section 4.6, is the second contribution of this direction of work.

Given an original ARS, the proof is based on the construction of a second ARS, coinciding in objects, steps and residuals with the original one, and whose embedding relation is a total order including the embedding of the original ARS. The proof states that standardisation existence for the original ARS, together with standardisation uniqueness for the second ARS, imply standardisation uniqueness for the original ARS.

We also remark that the definition of an ARS modeling $\lambda_{\mathtt{lsub}}^{\sim}$ imposes two challenges.

Firstly, to define the *embedding* relation describing the power of a step to duplicate or erase others, since such embedding does not coincide with nesting between the corresponding redexes.

Secondly, to obtain definitions of steps, residuals and embedding being *stable* by the equivalence relation on terms, generated by the equations commented in Section 1.2.2. We obtain stable definitions by *labeling* each redex in a term, analogously as the use of labels in Fig. 1.1. We subsequently adapt the equivalence equations to labeled terms, and equate redexes conveying the same label in equivalent labeled terms. Stability is proved by observing some invariants about labels w.r.t. application of the equivalence equations on labeled terms.

1.4.3 Equivalence of reductions for infinitary rewriting systems

As we mentioned, various results about some basic properties (as termination, confluence, or uniqueness of normal forms) for infinitary term rewriting systems are present in the literature, cfr. [Ken92, KKSdV95, BKdV03, KdV05, Zan08, EGH$^+$10, EHK12]. On the other hand, the only characterisation of *equivalence* of reductions for such systems known by the author is the definition of equivalence given in [KKSdV95], based on that presented in [HL91].

The first contribution in this direction is a new **characterisation of equivalence for strongly convergent reductions in infinitary, left-linear, first-order term**

rewriting systems. It is based on the notion of *step permutation*: for any pair of orthogonal rewrite steps, say a and b, perform a followed by b (more precisely, by the residuals of b after a), is equivalent to perform b followed by (the residuals of) a (after b).

This characterisation allows to verify the equivalence of infinite reduction sequences, operating on infinite terms, in several examples we analysed. Particularly, it allows to model the permutation of a step w.r.t. an infinite number of steps. Consider the rules $f(x) \to g(x)$ and $m(x) \to n(x)$, and let us use f^ω to denote the term $f(f(f(\ldots$. In order to verify the equivalence between the following two reductions:

$$\begin{array}{l} m(f^\omega) \to m(g(f^\omega)) \to m(g(g(f^\omega))) \twoheadrightarrow m(g^\omega) \to n(g^\omega) \\ m(f^\omega) \to n(f^\omega) \to n(g(f^\omega)) \to n(g(g(f^\omega))) \twoheadrightarrow n(g^\omega) \end{array}$$

the last step in the former reduction must be permuted with an infinite number of steps, since it corresponds with the first step in the latter reduction. The obtained characterisation also allows to model an *infinite number of step permutations*.

The obtained equivalence characterisation also allows to describe adequately the phenomenon, unique to infinitary rewriting, of *infinitary erasing*.

We use the *proof term* model to study infinitary rewriting. We give in this thesis an **extension of the proof term model**, as it is presented in [BKdV03] Sec. 8.2 and 8.5 for finitary, left-linear, first-order term rewriting, **to infinitary term rewriting**. This is a second contribution of the work in this direction. This extension is also limited to *left-linear* rewriting systems.

The obtained model is *complete*: we prove that any infinitary reduction sequence can be represented by a proof term. Moreover, the representation is proven *unique* up to (an infinitary extension of) rebracketing.

We remark that the definition of the set of proof terms is given by *inductive*, rather than coinductive, means. Transfinite induction is used to reason on proof terms, the limit case being infinite concatenation chains.

The starting point to model infinitary reduction equivalence through proof terms, is the congruence generated by six basic equations, which is proposed in [BKdV03] to chraracterise finitary reduction equivalence. We extended this definition in two ways: besides adding a new basic equation, we propose two novel *congruence rules*. One of the new rules incorporates the notion of *limit* into the equivalence judgements: if the difference between two reductions can be made as less relevant as desired, then the two reductions can be considered as equivalent. Relevance of a reduction is measured by the (inverse of the depth of the) unaffected prefix. E.g., a reduction including a head step[17] has the greatest relevance, since there is no unaffected prefix in this case. Such a relevance criterion is in line with the notion of strong convergence.

We also present a **novel proof of the compression result**, which is the third contribution in this direction. The proof is based on the equivalence of infinitary reductions, using our definition. We remark that it is the first proof, at the extent of the author's knowledge, which applies to both orthogonal and non-orthogonal rewriting

[17] that is, a step which involves the head symbol of a term. If we consider the rule $f(x) \to g(x)$, then the step $f(f(a)) \to g(f(a))$ is a *head step*, while this is not the case for $f(f(a)) \to f(g(a))$. In the latter example, the unaffected prefix corresponds to the outer occurrence of f, which is the *head symbol* of both $f(f(a))$ and $f(g(a))$.

systems, and at the same time asserts that the compressed reduction is equivalent to the original one.

1.4.4 Previous presentations of the results

Material included in this thesis has been presented in different conferences, as we detail in the following:

1. The strategy defined for PPC, and the normalisation proof in the version valid for PPC only, was presented in RTA 2012, cfr. [BKLR12]. This work was developed in collaboration with Delia Kesner, Eduardo Bonelli and Alejandro Ríos.

2. The abstract normalisation proof described in Section 3.3, which was developed in collaboration with Delia Kesner, Eduardo Bonelli and Alejandro Ríos, is the subject of an article in preparation.

3. The results about $\lambda_{\mathtt{lsub}}^{\sim}$ were presented in POPL 2014, cfr. [ABKL14]. This work was developed in collaboration with Delia Kesner, Beniamino Accattoli and Eduardo Bonelli.

4. The results about infinitary rewriting were presented in RTA 2014, cfr. [LRdV14].

   Work in this direction was firstly presented at the first Workshop on Infinitary Rewriting held in 2013, cfr. `http://joerg.endrullis.de/wir.html`.

   This work was developed in collaboration with Alejandro Ríos and Roel de Vrijer.

Chapter 2

Preliminaries – generic models of rewriting systems

We describe in this chapter the generic models of rewriting systems we use in this thesis: the *Abstract Rewriting Systems*, in the formulation of [Mel96], and the *proof term model*, as presented for first-order term rewriting systems in [BKdV03]. In the introduction, these models are presented in Sections 1.3.1 and 1.3.2 respectively.

2.1 Abstract Rewriting Systems

The main elements and features of the ARS model, as well as the ideas shaping this model, are described in Section 1.3.1. In the following, we formalise the definition of an ARS, recall some notations presented in Section 1.3.1, introduce some new notations and notions, and give additional examples. Afterwards, we present the axioms which formalise the features of a rewriting system, when modeled as an ARS. A short description of the intent of each axiom is included. Subsequently, we describe how equivalence of reductions and standardisation are captured in the ARS model. A brief comment about total embeddings closes this presentation.

We follow the presentation of the ARS model given in [Mel96].

2.1.1 Basic elements

The basic definition of an ARS follows.

Definition 2.1.1 (ARS with embedding)**.** *An* ***ARS with embedding*** *is defined as a tuple* $\langle \mathcal{O}, \mathcal{R}, \mathsf{src}, \mathsf{tgt}, [\![\cdot]\!], < \rangle$ *where* $\mathcal{O}$ *and* $\mathcal{R}$ *are the sets of* ***objects*** *and* ***steps*** *respectively,* $\mathsf{src}, \mathsf{tgt} : \mathcal{R} \to \mathcal{O}$ *are the* ***source*** *and* ***target*** *functions,* $[\![\cdot]\!] \subseteq \mathcal{R} \times \mathcal{R} \times \mathcal{R}$ *is the* ***residual*** *relation, and* $< \subseteq \mathcal{R} \times \mathcal{R}$ *is the* ***embedding*** *relation. The embedding relation must be a well-founded order.*

All these elements are described in Section 1.3.1. The work about normalisation developed in Chapter 3 uses an extension of the ARS model, which includes the *gripping* relation.

Definition 2.1.2 (ARS with embedding and gripping)**.** *An* ***ARS with embedding and gripping*** *is a tuple* $\langle \mathcal{O}, \mathcal{R}, \mathsf{src}, \mathsf{tgt}, [\![\cdot]\!], <, \ll \rangle$*, where* $\ll \subseteq \mathcal{R} \times \mathcal{R}$ *is the* ***gripping*** *relation; cfr. Section 1.3.1 and Section 3.3.1.*

Some notational conventions and basic definitions about the ARS model follow.

Notation 2.1.3 (ARS, elements of an ARS)**.** *We use $\mathfrak{A}, \mathfrak{A}', \mathfrak{A}_1$, etc., to denote an ARS. We usually use in this thesis the symbols $a, a', a_1, b, c, \ldots$ for steps, and $t, t', u, s, v, r, \ldots$ for objects. Another usual notation is $t \xrightarrow{a} u$, which denotes that $\mathsf{src}(a) = t$ and $\mathsf{tgt}(a) = u$. Recall from Section 1.3.1 that we use $b[\![a]\!]b'$ for $(b, a, b') \in [\![\cdot]\!]$, and that we adopt the infix notation for $<$ and $\ll$. Moreover, we use $b[\![a]\!]$ to denote the set $\{b' \ / \ b[\![a]\!]b'\}$.*

Notation 2.1.4 (Steps of an object)**.** *Given an object t, we write $\mathcal{RO}(t)$ to denote the set $\{a \ / \ \mathsf{src}(a) = t\}$.*

If $t \xrightarrow{a} u$ and $b' \in \mathcal{RO}(u)$, then it could be the case that b' is not the residual of any step in t. In this case, we say that b' is **created** by a.

Definition 2.1.5 (created step)**.** *Let $t \xrightarrow{a} u$ and $b' \in \mathcal{RO}(u)$, such that there is no $b \in \mathcal{RO}(t)$ verifying $b[\![a]\!]b'$. In this case, we say that b' is **created** by a, and we write $\varnothing[\![a]\!]b'$.*

Definition 2.1.6 (Coinitial steps)**.** *Two steps a and b are said **coinitial** iff $\mathsf{src}(a) = \mathsf{src}(b)$; the notion of **coinitial set of steps** is defined analogously. We use $\mathcal{A}, \mathcal{A}', \mathcal{B}, \mathcal{C}, \ldots$ to denote sets of coinitial steps.*

Definition 2.1.7 (Disjoint steps)**.** *If a and b are coinitial, and none of $a = b$, $a < b$ and $b < a$ hold, then we say that a and b are **disjoint** steps, notation $a \parallel b$.*

Definition 2.1.8 (Normal form)**.** *Let t be an object. If $\mathcal{RO}(t) = \varnothing$, then we say that t is a **normal form**. We denote the set of normal forms of an ARS by **NF**.*

The notion of *residual* can be extended to sets of coinitial steps.

Definition 2.1.9. *Let $\mathcal{B}$ be a set of coinitial steps, a a step, coinitial in turn with $\mathcal{B}$, and $b' \in \mathcal{RO}(\mathsf{tgt}(a))$. We say that b' is a **residual of the set of coinitial steps** $\mathcal{B}$ after a, notation $\mathcal{B}[\![a]\!]b'$, iff $b[\![a]\!]b'$ for some $b \in \mathcal{B}$. We also use the notation $\mathcal{B}[\![a]\!]$, to denote $\{b' \ / \ \mathcal{B}[\![a]\!]b'\}$.*

Notice that for any a and b, $b[\![a]\!]$ is a set of coinitial steps; the same happens with $\mathcal{B}[\![a]\!]$ for any $\mathcal{B}$.

The residual, embedding and gripping relations must verify the following condition: whenever $b[\![a]\!]b'$, $a < b$ or $a \ll b$, a and b must be coinitial; for the residual relation, $\mathsf{src}(b') = \mathsf{tgt}(a)$ is also required.

Recall that the intent of the residual relation is to trace a step b, after the contraction of a step a, so that $b[\![a]\!]b'$ indicates that b' is (perhaps part of) what is left of b, after a has been performed. Hence the restrictions $\mathsf{src}(b) = \mathsf{src}(a)$ and $\mathsf{src}(b') = \mathsf{tgt}(a)$ make sense. Cfr. Fig. 2.1.

$$\begin{array}{ccc} \underbrace{\mathsf{src}(a)}_{b \text{ is here}} & \xrightarrow{a} & \underbrace{\mathsf{tgt}(a)}_{b' \text{ is here}} \\ b & [\![a]\!] & b' \end{array}$$

Figure 2.1: Schema of the residual relation

Let us describe how the λ-calculus can be modeled as an ARS.

The set $\mathcal{O}$ of *objects* is the set of terms of the λ-calculus. A *step* corresponds to any subterm of the form $(\lambda x.s)u$, lying inside a term. The formal definition of the set of steps can be given by resorting to the concept of *position*, as done for `PPC` in Section 3.4.3, or alternatively by using *contexts*, as described for $\lambda_{\widetilde{\texttt{lsub}}}$ in Section 4.2.

W.r.t. the *residual relation*, $b[\![a]\!]b'$ holds iff b' is a "copy" of the (subterm corresponding to) the step b, in the target term of a. A graphical way of computing the residuals of b is to underline, in the common source term of a and b, the λ symbol of the subterm $(\lambda x.s)u$ corresponding to b, and perform the step a on the underlined term. Then $b[\![a]\!]b'$ holds iff the λ symbol of (the subterm corresponding to) b' has an underline, in the underlined version of $\texttt{tgt}(a)$. A similar technique, based on *labels*, is used for $\lambda_{\widetilde{\texttt{lsub}}}$ in Chapter 4. Cfr. Section 4.1.1. [1]

Fig. 2.2 includes several cases of steps and residuals. We use underlining to trace residuals. Cfr. also Fig. 1.6, on page 16.

a) $$((\overbrace{(\lambda x.x)3}^{a})\,(\underbrace{(\underline{\lambda} y.y)4}_{b}) \xrightarrow{a} 3\,(\underbrace{(\underline{\lambda} y.y)4}_{b'})$$

b) $$\underbrace{(\underline{\lambda} y.\,\overbrace{(\lambda x.x)3}^{a} + y)4}_{b} \xrightarrow{a} \underbrace{(\underline{\lambda} y.3+y)4}_{b'}$$

c) $$\overbrace{(\lambda x.\,\underbrace{(\underline{\lambda} y.x+x+y)3}_{b})4}^{a} \xrightarrow{a} \underbrace{(\underline{\lambda} y.4+4+y)3}_{b'}$$

d) $$\overbrace{(\lambda x.x+x+4)(\underbrace{(\underline{\lambda} y.y)3}_{b})}^{a} \xrightarrow{a} (\underbrace{(\underline{\lambda} y.y)3}_{b'}) + (\underbrace{(\underline{\lambda} y.y)3}_{b''}) + 4$$

e) $$\overbrace{(\lambda x.4)(\underbrace{(\underline{\lambda} y.y)3}_{b})}^{a} \xrightarrow{a} 4$$

f) $$\overbrace{(\lambda x.(x3)+4)(\lambda y.y)}^{a} \xrightarrow{a} ((\lambda y.y)3)+4$$

Figure 2.2: Examples of steps, residuals, and a created step, in the λ-calculus

In Fig. 2.2:a) to d), we have $b[\![a]\!]b'$, and moreover $b[\![a]\!]b''$ in d). The example c) shows that the subterm corresponding to a residual can be different from that of the original step. The example d) shows a case of **duplication**: there is more than one residual of b after a. The example e) shows a case of **erasure**: there is no residual of b after a. Finally, the example f) shows a case of **step creation**: the step in the target term, whose corresponding subterm is $(\lambda y.y)3$, is not the residual of any step in the source term.

The *embedding relation* can be defined as follows: $a < b$ if the subterm corresponding to b is nested inside that of a in their common source term. That is, a step corresponding

[1] The definition of the residual relation for `PPC` given in Chapter 3 is based on computations performed on the *positions* of steps. Cfr. Section 3.4.3.

to some subterm $(\lambda x.s)u$ inside a term, embeds those steps whose subterm is inside s or u. E.g., in Fig. 2.2:c), d), e), we have $a < b$, while in b) we have $b < a$.

As commented in Section 1.3.1, this is not the only possible definition of the embedding relation for λ-calculus. A meaningful alternative is to consider that the step whose subterm is $(\lambda x.s)u$ embeds only the steps whose subterms are inside u. Using the alternative definition, we have $a < b$ in Fig. 2.2:d) and e), but not in c).

Following the idea described in Section 1.3.1, we define the *gripping relation* for the ARS modeling the λ-calculus as follows: if the subterm for a step a is $(\lambda x.s)u$, then we have $a \ll b$ iff the subterm for b is inside s and x occurs free in that subterm. E.g., in Fig. 2.2:c) we have $a \ll b$, because x occurs free in $(\lambda y.x + x + y)3$. On the other hand, in b) we have $b < a$ but not $b \ll a$, since y does not occur free in $(\lambda x.x)3$.

2.1.2 Reduction sequences and developments

Sequences of rewriting steps admit a natural description in the ARS model.

Definition 2.1.10 (Reduction sequence, source, target, length)**.** *A* ***reduction sequence*** *is either* $\mathtt{nil}_t$*, an* empty sequence *indexed by the object* t*, or a (possibly infinite) sequence* $a_1; a_2, \ldots; a_n; \ldots$ *of steps verifying* $\mathsf{tgt}(a_k) = \mathsf{src}(a_{k+1})$ *for all* $k \geqslant 1$*. In the former case, we define the* ***source*** *as* t *and in the latter case as the source of the first step in the sequence. We define the* ***target*** *of a reduction sequence as follows:* $\mathsf{tgt}(\mathtt{nil}_t) := t$*,* $\mathsf{tgt}(a_1; \ldots; a_n) := \mathsf{tgt}(a_n)$*. The* ***length*** *of a reduction sequence, denoted by* $|\cdot|$*, is defined as follows:* $|\mathtt{nil}_t| := 0$*,* $|a_1; \ldots; a_n| := n$*. The target and length of an infinite sequence are undefined.*

Some notations about reduction sequences follow.

Notation 2.1.11. *We write* $\mathcal{RS}$ *for the set of reduction sequences. In the following, reduction sequences are given the names* δ*,* δ'*,* δ_1*,* γ*,* π*, etc. We write* $t \xrightarrow{\delta}\!\!\!\!\rightarrow u$ *to indicate that* $\mathsf{src}(\delta) = t$ *and* $\mathsf{tgt}(\delta) = u$*. Also, if* $\delta = a_1; \ldots; a_n$*, we denote with* $\delta[k]$ *the step* a_k*, and write* $\delta[i..j]$ *for the subsequence* $a_i; \ldots; a_j$*, if* $i \leqslant j$*, and* $\mathtt{nil}_{\mathsf{src}(a_i)}$*, if* $i > j$*. We use the symbol* ; *to denote the concatenation of reduction sequences, allowing to concatenate steps and sequences freely, e.g.* $a; \delta$ *or* $a; b$ *or* $\delta; a$ *or* $\delta; \gamma$*, as long as the concatenation yields a valid reduction sequence.*

The concept of **normalising** object is crucial for Chapter 3.

Definition 2.1.12 (Normalising object)**.** *An object* t *is* ***normalising*** *iff there exists a reduction sequence* δ *such that* $t \xrightarrow{\delta}\!\!\!\!\rightarrow u$ *and* u *is a normal form.*

The notion of *residuals* can be extended to reduction sequences as follows.

Definition 2.1.13 (Residuals after a reduction sequence)**.** *The relation of* ***residuals of a step after a reduction sequence****,* $[\![\cdot]\!] \subseteq \mathcal{R} \times \mathcal{RS} \times \mathcal{R}$*, is defined as follows:* $b[\![\mathtt{nil}_t]\!]b$ *for all* $b \in \mathcal{RO}(t)$*, and* $b[\![a;\delta]\!]b'$ *whenever* $b[\![a]\!]b''$ *and* $b''[\![\delta]\!]b'$ *for some* b''*. We use the notation* $b[\![\delta]\!]$*, to denote* $\{b' \ / \ b[\![\delta]\!]b'\}$*. We extend the definition of residuals after a reduction sequence to* sets of coinitial steps*, as follows: we say that* $\mathcal{B}[\![\delta]\!]b'$ *iff* $b[\![\delta]\!]b'$ *for some* $b \in \mathcal{B}$*, and define* $\mathcal{B}[\![\delta]\!]$ *as* $\{b' \ / \ \mathcal{B}[\![\delta]\!]b'\}$*; cfr. Dfn. 2.1.9. Observe that* $\mathcal{B}[\![a;\delta]\!] = \mathcal{B}[\![a]\!][\![\delta]\!]$*.*

The central role that residuals play in the ARS model yields a natural way to describe *developments*.

Definition 2.1.14 (Development, complete development)**.** *Let $\mathcal{A} \subseteq \mathcal{RO}(t)$ for some object t. The reduction sequence δ is a* ***development*** *of $\mathcal{A}$ iff $\mathsf{src}(\delta) = t$ and $\delta[i] \in \mathcal{A}[\![\delta[1..i-1]]\!]$ for all $i \leqslant |\delta|$ (the condition $\mathsf{src}(\delta) = t$ is in fact redundant unless $\mathcal{A} = \varnothing_t$). A development δ of $\mathcal{A}$ is* ***complete****, written $\delta \Vdash \mathcal{A}$, iff δ is finite and $\mathcal{A}[\![\delta]\!] = \varnothing$.*

E.g. let us consider the set $\mathcal{A} = \{a, b\} \subseteq \mathcal{RO}(t)$ given by

$$t = \overbrace{(\lambda x.(\lambda z.3 + z)x)(\underbrace{I4}_{b})}^{a}$$

The reduction sequence $(\lambda x.(\lambda z.3 + z)x)(I4) \xrightarrow{a} (\lambda z.3 + z)(I4) \xrightarrow{b'} (\lambda z.3 + z)4$ is a development of $\mathcal{A}$: observe that $a \in \mathcal{A} = \mathcal{A}[\![\mathtt{nil}_t]\!]$, and $b[\![a]\!]b'$ implies $b' \in \mathcal{A}[\![a]\!]$. The reduction sequence $(\lambda x.(\lambda z.3 + z)x)(I4) \xrightarrow{b} (\lambda x.(\lambda z.3 + z)x)4 \xrightarrow{a'} (\lambda z.3 + z)4$, where $a[\![b]\!]a'$, is also a development of $\mathcal{A}$. Note also that the reduction sequence consisting solely of the step a is a development of $\mathcal{A}$ too, and analogously for the step b. On the other hand, $(\lambda x.(\lambda z.3 + z)x)(I4) \xrightarrow{a} (\lambda z.3 + z)(I4) \to 3 + (I4)$ is not a development of $\mathcal{A}$, because the second step is not in $\mathcal{A}[\![a]\!]$.

Developments yield a useful measure on sets of coinitial steps.

Definition 2.1.15 (Depth of a set of coinitial steps)**.** *The* ***depth*** *of a set of coinitial steps $\mathcal{A}$, written $\nu(\mathcal{A})$, is the length of its longest complete development.*

Note that it is not a priori clear that a development terminates, nor that the residual relation is finitely branching. Moreover, since there may be more than one development of a given set of coinitial steps, it is natural to wonder whether they all have the same target and induce the same residual relation. These topics are discussed when introducing the finite residuals, finite developments and semantic orthogonality axioms.

2.1.3 Initial axioms

The ARS model allows to state and prove properties in an abstract fashion. The results thus obtained are valid for any ARS (and therefore, for any rewriting system which can be modeled as an ARS), provided that it verifies some properties. These requirements are encoded in the ARS model as *axioms*, stated in an abstract way.

Several of the axioms introduced in [Mel96] are used in this thesis. They are described in this and the following sections. In this section we describe three initial axioms. Later sections deal with the *finite developments* and the *semantic orthogonality* axioms, the group of *embedding* axioms, concerning the interaction between residuals and embedding, and finally the group of *gripping* axioms, which express basic properties of the gripping relation. We remark that the material in Chapter 3 requires an additional axiom, not present in [Mel96]; cfr. Section 3.1.4.

A note about notation: in what follows, free variables in the statement of an axiom are implicitly assumed as universally quantified. For example, "$a[\![a]\!] = \varnothing$" should be read as "For all $a \in \mathcal{R}$, $a[\![a]\!] = \varnothing$". Bear in mind also that in an expression such as "$a[\![b]\!]a'$", steps a and b are assumed coinitial.

The three *initial axioms* have to do with the properties of the residual relation. The embedding and gripping relations do not participate in these axioms. The first is *Self Reduction* and states, quite reasonably, that nothing is left of a step a if it is contracted.

Self Reduction $a[\![a]\!] = \varnothing$.

The second is *Finite Residuals* and states that the residuals of a step b after contraction of a coinitial (and possibly the same) one a is a finite set. In other words, a step may erase ($b[\![a]\!] = \varnothing$) or copy other coinitial steps, however only a finite number of copies can be produced.

Finite Residuals $b[\![a]\!]$ is a finite set.

The third one, namely *Ancestor Uniqueness*, states that a step a cannot "fuse" two different steps b_1 and b_2, coinitial with a, into one, by allowing some step b to be residual of both b_1 and b_2 simultaneously. In other words, if we use the term "ancestor" to refer to the inverse of the residual relation, then any step can have at most one ancestor (recall that *created* steps have no ancestor).

Ancestor Uniqueness $b_1[\![a]\!]b'$ and $b_2[\![a]\!]b' \Rightarrow b_1 = b_2$.

2.1.4 Finiteness of developments and semantic orthogonality

As indicated at the end of Section 2.1.2, it is not clear, in principle, whether *developments* enjoy certain desired properties. In this section we address this problem by introducing two axioms which guarantee the expected behavior of developments.

The *finite developments* axiom, acronym FD, asserts that no development can run indefinitely.

Finite developments (FD) All developments of $\mathcal{A}$ are finite.

This axiom, together with Finite Residuals, imply that the notion of *depth of a set of coinitial steps*, cfr. Section 2.1.2, is well-defined.[2]

In turn, an additional axiom, called PERM in [Mel96] and *semantic orthogonality*, acronym SO, in this thesis, guarantees, for any pair of coinitial steps $\mathcal{A} = \{a, b\}$, the existence of two complete developments of $\mathcal{A}$, one starting with a and the other with b, which are confluent and induce the same residual relation. Cfr. Fig. 2.3.

Semantic orthogonality (SO) $\exists \delta, \gamma.\ \delta \Vdash a[\![b]\!]$ and $\gamma \Vdash b[\![a]\!]$ and $\mathsf{tgt}(a;\gamma) = \mathsf{tgt}(b;\delta)$ and the relations $[\![a;\gamma]\!]$ and $[\![b;\delta]\!]$ coincide.

[2]If we render the developments of a set of coinitial steps $\mathcal{A}$ as a tree, whose root is the source term of $\mathcal{A}$ and each edge is a step, then the Finite Residuals axiom implies that such tree is finitely branching, and FD entails the nonexistence of infinite branches. Therefore König's Lemma yields that the described tree is finite, hence the well-definedness of the depth of $\mathcal{A}$.

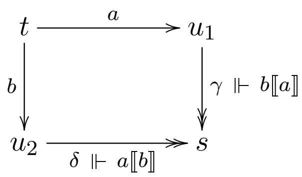


Figure 2.3: The semantic orthogonality axiom

The axioms FD and SO, together with Self Reduction and Finite Residuals, suffice to guarantee that complete developments of *arbitrary* multisteps of an ARS are also confluent and induce the same residual relation. This is reflected in the following result (Lem. 2.18 and Lem. 2.19 in [Mel96]):

Proposition 2.1.16. *Consider an ARS enjoying the Self Reduction, Finite Residuals, FD and SO axioms. Suppose $\delta \Vdash \mathcal{A}$ and $\gamma \Vdash \mathcal{A}$. Then $\mathsf{tgt}(\delta) = \mathsf{tgt}(\gamma)$ and the relations $[\![\delta]\!]$ and $[\![\gamma]\!]$ coincide.*

The properties expressed by the axioms FD and SO have long been present in the formal study of rewriting systems, allowing to obtain relevant results.

We mention the proof of confluence (property described in Section 1.1.1) given for a variant of the λ-calculus in [CR36], where FD is explicitly stated and proved. Early proofs of this axiom for λ-calculus can be found in [Sch65, Hin78].

In turn, *orthogonality* is a regularity criterion which simplifies the analysis of rewriting systems. The study of the so-called *orthogonal rewriting systems* can be traced, at least, to [HL91], which is in fact a revised version of a technical report from 1979. This work, whose subject is first-order term rewriting, takes a *syntactic* approach to orthogonality, based on the notion of *ambiguity*.[3] We describe this notion by means of an example. The inclusion, in a first-order term rewriting system, of the rules

$$h(f(x), y) \to x \qquad\qquad h(x, g(y)) \to y$$

provokes an ambiguity w.r.t. all the terms having the form $h(f(t_1), g(t_2))$ where t_1 and t_2 are arbitrary terms, since both rules apply to any such term. Taking as example the term $h(f(c), g(d))$, we have two steps a_1 and a_2, where $h(f(c), g(d)) \xrightarrow{a_1} c$ and $h(f(c), g(d)) \xrightarrow{a_2} d$. The absence of ambiguities is a requirement for a rewriting system to be orthogonal.[4]

As the example suggests, lack of orthogonality can break confluence, and thus uniqueness of normal forms. In the syntactic view of orthogonality, the statement of the SO axiom is in fact a property, which is proved for any (syntactically) orthogonal rewriting system, cfr. Prop. 4.2.8 and Prop. 4.2.10 in [BKdV03], page 96.[5] Besides Chapter 4 in [BKdV03], the syntactic approach of orthogonality is also described in [BN98], Section 6.3.

The ARS model takes a different, more *semantically*-oriented perspective on orthogonality. In this view, a rewriting system is *defined* as orthogonal iff it satisfies the criterion about the meeting of developments expressed by the SO axiom.

[3] The notions of *overlapping* [Ros73], and of *critical pair* [KB70], refer to the same phenomenon.

[4] There is another requirement, i.e., a rewriting system must be also *left-linear* to be considered orthogonal. Cfr. [HL91] p. 398, [BKdV03] p. 88.

[5] The statement of the SO axiom a stronger version of the *local confluence*, or *WCR*, property, which states that whenever $t \to u_1$ and $t \to u_2$, there exists an object s verifying $u_1 \twoheadrightarrow s$ and $u_2 \twoheadrightarrow s$.

We remark the existence of term rewriting systems which verifies the semantic orthogonality criterion described by the SO axiom, despite the fact that they admit syntactic ambiguities. Let us consider the first-order term rewriting system whose only rules are

$$h(f(x), y) \rightarrow e \qquad\qquad h(x, g(y)) \rightarrow e$$

Again, any term having the form $h(f(t_1), g(t_2))$ is ambiguous for this system. But in this case, the possible steps are $h(f(t_1), g(t_2)) \xrightarrow{a_1} e$ and $h(f(t_1), g(t_2)) \xrightarrow{a_2} e$, for any such term. Moreover, none of these steps has any residual after the other one. Hence semantic orthogonality is not compromised: the diagram in Fig. 2.3 closes trivially as $u_1 = u_2 = s = e$. In fact, we can *identify* the steps a_1 and a_2 in this case, considering the existence of just one step $h(f(t_1), g(t_2)) \xrightarrow{a} e$.

Another ambiguous system, non-orthogonal from a syntactic perspective, which enjoys the SO axiom, is the *parallel-or* first-order term rewriting system, referenced in the literature at least since [Plo77] in relation with denotational semantics, and [Ken89] specifically in relation with the existence of *normalising reduction strategies*. It includes the following rules

$$\texttt{or}(x, \texttt{tt}) \rightarrow \texttt{tt} \qquad\qquad \texttt{or}(\texttt{tt}, x) \rightarrow \texttt{tt}$$

Both rules apply to $\texttt{or}(\texttt{tt}, \texttt{tt})$, giving rise to two different steps, and therefore to an ambiguity. On the other hand, the target of both steps, namely tt, coincide. Therefore, the behavior of this system is analogous to the previous example, so that again we can identify the two ambiguous steps.

Observe that both of the just presented rewriting systems are *almost orthogonal*, cfr. [vR97], from a syntactic perspective. On the other hand, the linear substitution calculus we study in Chapter 4, is syntactically not almost orthogonal: moreover, it is not even *weakly orthogonal*, [vO94]; despite this fact, it verifies the SO axiom.

A further comment about the two perspectives on orthogonality is included in Section 6.2.3.

In the following, we say that an ARS is **orthogonal** iff it satisfies the three initial axioms, FD and SO.

2.1.5 Embedding axioms

The embedding axioms establish coherence conditions between the embedding relation $<$ and the residual relation $[\![\cdot]\!]$. In reading these axioms it helps to think about the embedding relation as described for the λ-calculus in Section 2.1.1: $a < b$ if the subterm corresponding to a nests that of b. Bear in mind however, that the ARS model does not assume the existence of terms nor of syntactic nesting; this reading is solely for the purposes of aiding the interpretation of the axioms.

The first axiom, Linearity, states that the only way in which a step a can either erase or produce multiple (two or more) copies of a coinitial step, is if it embeds it.

Linearity $\qquad a \not< b \Rightarrow \exists! b' \;/\; b[\![a]\!]b'.$

This axiom formalises the intent of the embedding relation as part of the ARS model, described in Section 1.3.1: a pair $a < b$ indicates that the step a has, potentially, the power to erase or to duplicate b.

The second axiom pertains to the invariance of the embedding relation w.r.t. contraction of steps. Consider three coinitial steps a, b and c. Suppose that $b[\![a]\!]b'$ and $c[\![a]\!]c'$, for some steps b' and c' (this implies $a \neq c$ and $a \neq b$). The only case in which the contraction of a can add, to (the residual of) b, the feature of embedding (that of) c, i.e. $b \not< c \wedge b' < c'$, or conversely, revoke this feature (in the residuals), that is $b < c \wedge b' \not< c'$, is when the step a itself embeds c, that is, when $a < c$.

Context-Freeness $\quad b[\![a]\!]b'$ and $c[\![a]\!]c' \Rightarrow a < c \vee (b < c \Leftrightarrow b' < c')$.

An example in λ-calculus follows; recall that I, K and D are defined at the end of Section 1.1.4:

$$\overbrace{(\lambda x.\underbrace{Dx}_{b})(\underbrace{I3}_{c})}^{a}(\underbrace{K\overbrace{I4}^{e}}_{d}) \xrightarrow{a} \underbrace{D(\underbrace{I3}_{c'})}_{b'}(\underbrace{K\overbrace{I4}^{e'}}_{d'})$$

In this case, $a < c$. Therefore, the axiom allows to modify the relative embedding of b and c after the contraction of a, as it is indeed the case: $b \not< c$ and $b' < c'$. On the other hand, $a \not< d$ and $a \not< e$, hence the relative embeddings of any step with d and e must be invariant w.r.t. the contraction of a. We observe e.g. $b \not< d$ and $b' \not< d'$, and also $d < e$ and $d' < e'$.

The next two axioms, Enclave–Creation and Enclave–Embedding, are used in contexts in which the axiom Linearity is assumed. Consider two coinitial steps a and b, such that $b < a$, so that Linearity implies the existence of a unique b' verifying $b[\![a]\!]b'$. The Enclave axioms establish conditions which guarantee, given some c' coinitial with b', that $b' < c'$. Two cases are considered, first when c' is created by a (Enclave–Creation), and when it is a residual, after a, of some step c (Enclave–Embedding).

Enclave–Creation $\quad b < a$, $b[\![a]\!]b'$ and $\varnothing[\![a]\!]c' \Rightarrow b' < c'$.
Enclave–Embedding $\quad b[\![a]\!]b'$, $c[\![a]\!]c'$ and $b < a < c \Rightarrow b' < c'$.

Notice that Enclave–Embedding complements, in some sense, Context-Freeness: it enforces the invariance of the relative embedding between b and c, that is $b < c \wedge b' < c'$, in a case where $a < c$, so that the case is not covered by Context-Freeness.

We illustrate the Enclave–Creation and Enclave–Embedding axioms by means of two examples in λ-calculus. Consider

$$a)\quad \overbrace{(\lambda x.\underbrace{Dx}_{d})(\underbrace{IK}_{a}\,3)}^{b} \xrightarrow{a} \overbrace{(\lambda x.\underbrace{Dx}_{d'})(\underbrace{K3}_{c'})}^{b'}$$

$$b)\quad \overbrace{(\lambda x.\underbrace{Dx}_{d})(\underbrace{I(\underbrace{K3}_{c})}_{a})}^{b} \xrightarrow{a} \overbrace{(\lambda x.\underbrace{Dx}_{d})(\underbrace{K3}_{c'})}^{b'}$$

In a), the step c' is created by the contraction of a. Moreover $b < a$, so that Enclave–Creation enforces $b' < c'$. On the other hand, $d \not< a$, so that this axiom does not assert anything about the relative embeddings of d' and c'. In b), we have $b < a < c$, so that Enclave–Embedding implies $b' < c'$.

An additional axiom, which complements in turn Context-Freeness and Enclave–Embedding, is introduced in Section 3.1.4.

The last embedding axiom in this presentation, *Stability*, assumes implicitly Linearity, as well as Ancestor Uniqueness and SO, to hold. Suppose two coinitial steps, a and b, such that $a \parallel b$, and let a', b' their unique mutual residuals, namely $a\llbracket b\rrbracket a'$ and $b\llbracket a\rrbracket b'$; Linearity implies the existence, and also the uniqueness, of a' and b'. Then SO implies that the target of $a; b'$ and $b; a'$ coincide. Let d' be a step in this common target, such that it is not created, neither by b' nor by a'. That is, $d_1\llbracket b'\rrbracket d'$ and $d_2\llbracket a'\rrbracket d'$, for some steps d_1 and d_2, coinitial with b' and a' respectively. Fig. 2.4:a) depicts this situation.

Assume the existence of a step d verifying $d\llbracket a\rrbracket d_1$, so that $d\llbracket a; b'\rrbracket d'$. In this case, SO implies $d\llbracket b; a'\rrbracket d'$, that is, $d\llbracket b\rrbracket d''$ and $d''\llbracket a'\rrbracket d'$ for some d''. In turn, Ancestor Uniqueness implies $d'' = d_2$, so that $d\llbracket b\rrbracket d_2$. Cfr. Fig. 2.4:b).

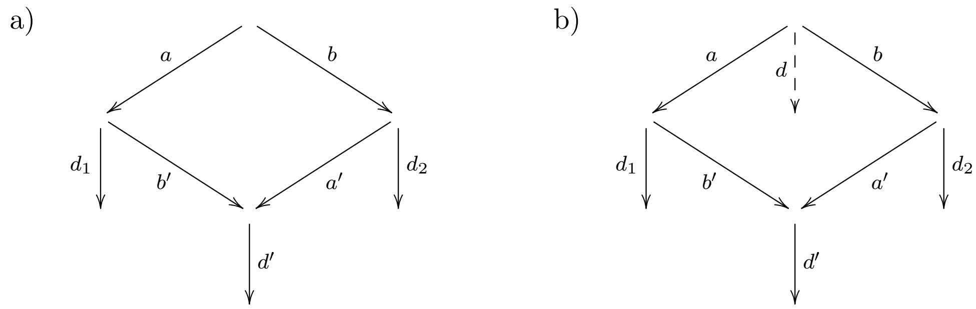


Figure 2.4: The Stability axiom

Therefore, there are just two options: either d_1 and d_2 are residuals of some common step d, or d_1 and d_2 are created by a and b respectively, so that both a and b have the ability to create (an ancestor of) d'. The axiom *Stability* forbids the latter possibility: it states that a step, in this case d', cannot be created by different, disjoint steps.

Stability Assume $a \parallel b$, $a\llbracket b\rrbracket a'$, $b\llbracket a\rrbracket b'$, and there exists some d' such that $d_1\llbracket b'\rrbracket d'$ and $d_2\llbracket a'\rrbracket d'$. Then there exists d such that $d\llbracket a\rrbracket d_1$, $d\llbracket b\rrbracket d_2$, and either $a \nleqslant d$ or $b \nleqslant d$.

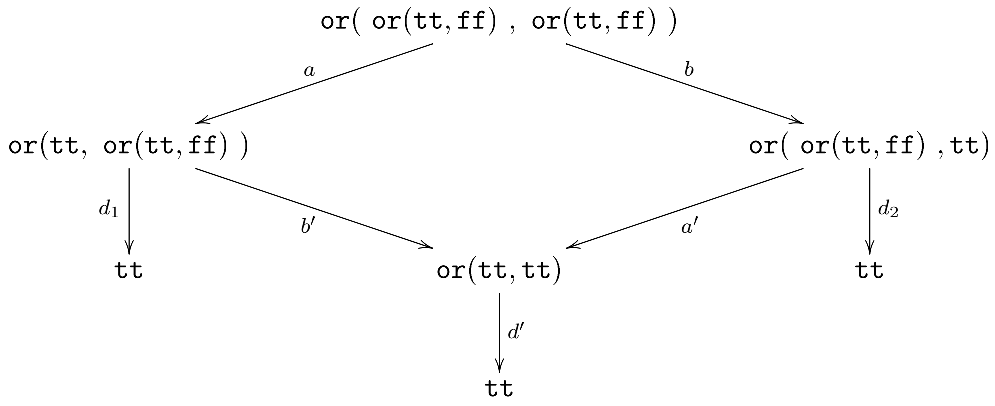
Therefore, in any case corresponding to Fig. 2.4:a), the situation in Fig. 2.4:b) must hold. In the latter we distinguish the conclusion of the axiom, that is, the existence of d, by a dashed line.

We point out that the condition $a \nleqslant d$ or $b \nleqslant d$ is superfluous for the λ-calculus, and also for all the ARS we introduce in Chapter 3 and Chapter 4, because for those ARS, this condition holds for every set of three coinitial steps $\{a, b, d\}$ such that $a \parallel b$. Notice that $d\llbracket a\rrbracket d_1$ and $d\llbracket b\rrbracket d_2$ imply $d \neq a$ and $d \neq d$, cfr. Self Reduction. Moreover, given an object t, the embedding relation restricted to $\mathcal{RO}(t)$ has the shape of a *tree*, so that $a < d$ and $b < d$ would imply $a < b$ or $b < a$, contradicting $a \parallel b$.

The *parallel-or* rewriting system, introduced in Section 2.1.4, does not enjoy Stability. Consider the term

$$\mathtt{or}(\ \underbrace{\mathtt{or}(\mathtt{tt}, \mathtt{ff})}_{a}\ ,\ \underbrace{\mathtt{or}(\mathtt{tt}, \mathtt{ff})}_{b}\ )$$

and the following diagram



If we identify the two possible steps $\texttt{or(tt,tt)} \to \texttt{tt}$ in the ARS interpretation of this rewriting system, then both a and b can create (an ancestor of) this "unified" step. Moreover $a \parallel b$. Hence this is a counterexample for Stability. This is by no means accidental: the explicit purpose of the Stability axiom is to avoid what is called in [Mel96] (cfr. page 80) the "*parallel-or* behavior".

2.1.6 Gripping axioms

The properties characterising the *gripping relation* in the ARS model, are described by means of three axioms, provided in [Mel96] to extend to higher-order term rewriting an abstract proof of finite developments, developed originally for first-order term rewriting systems by O'Donnell, cfr. [O'D77].

The first one, *Grip–Instantiation*, states the role gripping plays in the creation of new embeddings. Consider three coinitial steps a, b, c and steps b', c' such that $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Suppose that $b' < c'$, and moreover, that this embedding is generated by the contraction of a, that is, $b \not< c$. Axiom Context-Freeness gives some information, since it enforces $a < c$ in such case. This axiom may be seen to provide further information: the only way in which a can place (the residual of) c under the (residual of) b, is that b grips a.

Grip–Instantiation $\quad b[\![a]\!]b'$, $c[\![a]\!]c'$ and $b' < c' \Rightarrow b < c \vee (a \ll b \wedge a < c)$.

Recall the example for λ-calculus given in Section 1.3.1 to introduce gripping:

$$\overbrace{(\lambda x.\underbrace{Dx}_{b})(\overbrace{I3}^{c})}^{a} \xrightarrow{a} \underbrace{D(\overbrace{I3}^{c'})}_{b'}$$

We have $b \not< c$ and $b' < c'$. The new embedding is *generated* by the presence of a free occurrence of x, the variable bound in the abstraction corresponding to a, inside the subterm of b. This link between the steps a and b is exactly the phenomenon modeled by the gripping relation.

The second axiom, *Grip–Density*, states a condition for the generation of a new gripping. Consider again $b[\![a]\!]b'$ and $c[\![a]\!]c'$. The contraction of a can cause c' to grip b'

when this is not the case for their respective ancestors, i.e. $b \not\ll c$ and $b' \ll c'$, only if a links b and c forming a "chain" of grippings, that is, $b \ll a \ll c$: the contraction of a makes b' and c' contiguous in this chain.

Grip–Density $\qquad b[\![a]\!]b' \,\wedge\, c[\![a]\!]c' \,\wedge\, b' \ll c' \;\Rightarrow\; b \ll c \,\vee\, b \ll a \ll c.$

An example in λ-calculus follows: $\overbrace{(\lambda y.\,\underbrace{(\lambda x.\,\underbrace{Ix}_{c})y}_{a})\,z}^{b} \xrightarrow{a} \overbrace{(\lambda y.\,\underbrace{Iy}_{c'})\,z}^{b'}$

The third axiom, *Grip–Convexity*, establishes conditions to embed a gripping step: if c embeds b which in turn grips a, then c either grips or embeds a.

Grip–Convexity $\qquad a \ll b \,\wedge\, c < b \Rightarrow a \ll c \,\vee\, c \leqslant a.$

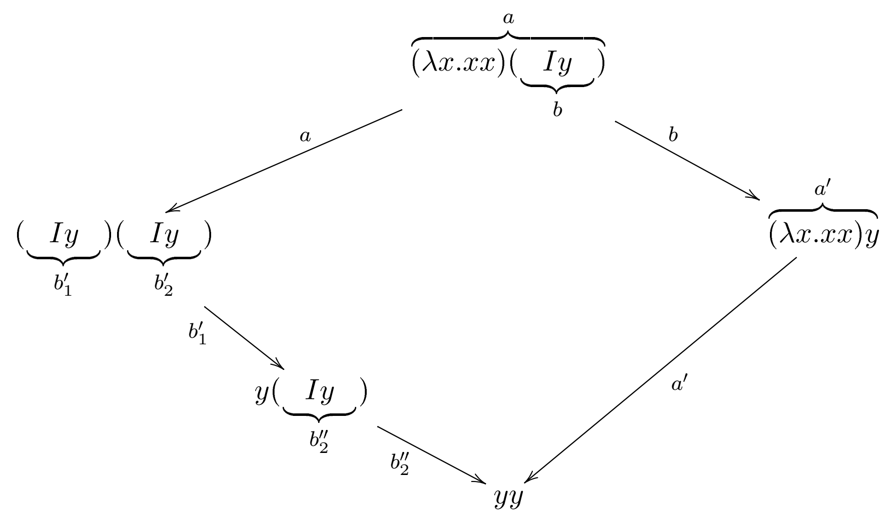
Consider this example in λ-calculus: $\overbrace{I(\underbrace{(\lambda x.\,\overbrace{I(\underbrace{Dx}_{b})}^{c})3}_{a})}^{d},$

where $a \ll b$, $c < b$, and also $d < b$. We have $a \ll c$ and $d < a$, so that both cases are compatible with the statement of Grip–Convexity.

2.1.7 Permutation equivalence in the ARS model

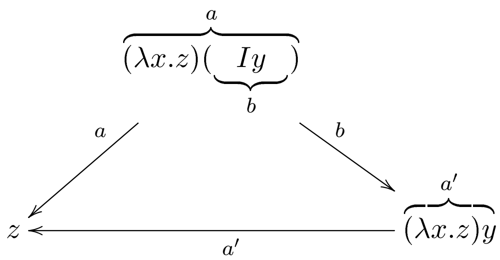
Residuals lead to a simple description, in the ARS model, of the *permutation of contiguous steps*. If a and b are coinitial steps, $\delta \Vdash b[\![a]\!]$, and $\gamma \Vdash a[\![b]\!]$, then the reduction sequence $a;\delta$ corresponds to the contraction of a followed by (the residuals of) b, while $b;\gamma$ corresponds to b followed by (the residuals of) a. Therefore, permuting a with b in $a;\delta$ yields $b;\gamma$, and vice versa.

This is the case of the example about equivalence of reductions given in Section 1.3.1 for the arithmetic simplification rewriting system. An example in the λ-calculus follows.



In this case, b has two residuals after a: $b[\![a]\!] = \{b'_1, b'_2\}$. Therefore, in this case, it takes more than one step to develop $b[\![a]\!]$, i.e. $|\delta| > 1$. On the other hand, $a[\![b]\!] = \{a'\}$. We have $\delta = b'_1; b''_2$ and $\gamma = a'$.[6] Permuting a with b in $a; b'_1; b''_2$ yields $b; a'$, and vice versa.

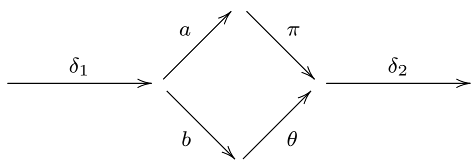
If we choose a and b such that a *erases* b, then we get a *triangular* diagram, e.g.:



In all the examples given so far, we obtain *closing* diagrams: the target term of $a; \delta$ and $b; \gamma$ coincide. This is always the case for *orthogonal* ARS, in a strong sense involving *residuals* as well as target: for any orthogonal ARS, and for any $\delta \Vdash b[\![a]\!]$ and $\gamma \Vdash a[\![b]\!]$, $\mathsf{tgt}(a; \delta) = \mathsf{tgt}(b; \gamma)$ and $c[\![a; \delta]\!] = c[\![b; \gamma]\!]$ for any c coinitial with a and b. This is a consequence of Prop. 2.1.16, since both $a; \delta$ and $b; \gamma$ are complete developments of $\{a, b\}$. Prop. 2.1.16 entails also that the choice of δ and γ is irrelevant, since target and residuals coincide for any complete development of $b[\![a]\!]$, and analogously for $a[\![b]\!]$.

The aforementioned considerations lead to the characterisation of permutation equivalence in the ARS model, for orthogonal ARS.

Definition 2.1.17. *Two reduction sequences δ and γ are* ***one permutation of steps away*** *if $\delta = \delta_1; a; \pi; \delta_2$, $\gamma = \delta_1; b; \theta; \delta_2$, $\pi \Vdash b[\![a]\!]$ and $\theta \Vdash a[\![b]\!]$. The permutation can be depicted graphically as follows.*



Definition 2.1.18. ***Permutation equivalence*** *is defined as the reflexive and transitive closure of the "one-permutation-away" relation.*[7]

Given Prop. 2.1.16, it is straightforward to verify that δ and γ being permutation equivalent implies $\mathsf{tgt}(\delta) = \mathsf{tgt}(\gamma)$ and $[\![\delta]\!] = [\![\gamma]\!]$.

2.1.8 Standardisation in the ARS model

Recall from Section 1.1.3 that in a *standard* reduction sequence, *external* steps should precede (residuals of) internal ones. The embedding relation allows to describe the

[6] Notice that a has the power of duplicating b. Therefore, a model of the λ-calculus as an ARS should provide $a < b$; cfr. the Linearity axiom. This is the case for the three embedding relations for the λ-calculus proposed in [Mel96], Section 2.7.2.

[7] Recall that Self Reduction implies, for any step a, that $a[\![a]\!] = \varnothing$. Therefore, if we consider $\delta = \delta_1; a; \mathtt{nil}_{\mathsf{tgt}(a)}; \delta_2$ and $b = a$ in Dfn. 2.1.17, it is easy to conclude that the relation of being "one permutation of steps away" is already reflexive, *except for empty reduction sequences.* Therefore, taking the reflexive closure in Dfn. 2.1.18 is needed for empty reduction sequences only. We could ask $a \neq b$ in Dfn. 2.1.17, in this case the reflexive closure in Dfn. 2.1.18 would be needed for any reduction sequence.

notion of "more external step" in the ARS model. Given two coinitial steps a and b, the condition $a < b$ indicates precisely that a is more external than b.

In this way, the external condition corresponds with the notion of *a step having some power over another*, introduced in Section 1.3.1: a step a is more external than b if a can possibly erase or duplicate b. The Linearity axiom makes this correspondence explicit.

Therefore, the criterion for a reduction sequence to be standard can be rephrased as follows: in a `s.r.s.`, a step a should precede (any residual of) a coinitial step b if a has some power on b.

In the following, assume an orthogonal ARS which enjoys also the Ancestor Uniqueness and Linearity axioms.

Consider a reduction sequence $\gamma = \delta_1; b; a'; \delta_2$ where $a[\![b]\!]a'$ and $a < b$. The presence of the **anti-standard pair** $b; a'$ indicates that γ is not a `s.r.s.`. Moreover, performing a permutation of the contiguous steps b and a on γ allows to "reorder" the anti-standard pair, obtaining $\delta_1; a; \pi; \delta_2$ where $\pi \Vdash b[\![a]\!]$. This observation leads to the following definition:

Definition 2.1.19 (Standardising permutation). *δ is obtained from γ by means of a **standardising permutation** (of contiguous steps), notation $\delta \angle \gamma$, iff $\gamma = \delta_1; b; a'; \delta_2$, $\delta = \delta_1; a; \pi; \delta_2$, $a[\![b]\!]a'$, $\pi \Vdash b[\![a]\!]$, and $a < b$.*

Recall that Linearity implies $a' \Vdash a[\![b]\!]$, hence a standardising permutation is indeed a particular case of the permutation of contiguous steps. Standardising permutations induce an order[8] on reduction sequences, as suggested by the symbol $\angle$ used to denote them.

Notice that permutations of *disjoint steps* do not affect the "standardisation degree" of a reduction sequence: they are *neutral* in that sense. On the other hand, performing such permutations can be required to unveil anti-standard pairs, and thus enable standardising permutations. This implies the relevance of the following definitions.

Definition 2.1.20 (Square permutation, square equivalence). *We say that δ and γ are **one square permutation** (of contiguous steps) **away**, notation $\delta \overset{1}{\diamond} \gamma$, iff $\delta = \delta_1; a; b'; \delta_2$, $\gamma = \delta_1; b; a'; \delta_2$, $b[\![a]\!]b'$, $a[\![b]\!]a'$, and $a \parallel b$. Linearity entails $b' \Vdash b[\![a]\!]$ and $a' \Vdash a[\![b]\!]$, hence a square permutation is indeed a permutation. We define the **square equivalence**, notation $\diamond$, as the reflexive and transitive closure of $\overset{1}{\diamond}$. It is immediate to verify that $\diamond$ is symmetric, and thus that it is indeed a equivalence relation.*

We show how to **standardise** a reduction sequence δ, i.e. obtain a `s.r.s.` permutation equivalent to δ, by performing square and standardising permutations. In the following, we (ab)use the same name for a step and its residuals. Consider:

$$\delta = \overbrace{(\lambda x.1x)(\underbrace{I2}_{b})}^{a}(\underbrace{I3}_{c}) \xrightarrow{b} \overbrace{(\lambda x.1x)2}^{a}(\underbrace{I3}_{c}) \xrightarrow{c} \overbrace{(\lambda x.1x)2}^{a}3 \xrightarrow{a} 123$$

Observe that δ is not standard because (the residual of) a comes after b, while $a < b$. Nonetheless, δ does not include any contiguous anti-standard pair, a *square permutation*

[8] A preorder in the general case, an order in the rewriting systems studied in this thesis

is needed prior to perform a standardising permutation. Namely, δ is transformed first to:

$$\delta' = \overbrace{(\lambda x.1x)(\underbrace{\ I2\ }_{b})}^{a}(\underbrace{\ I3\ }_{c}) \xrightarrow{b} \overbrace{(\lambda x.1x)2}^{a}(\underbrace{\ I3\ }_{c}) \xrightarrow{a} 12(\underbrace{\ I3\ }_{c}) \xrightarrow{c} 123$$

a reduction sequence including the anti-standard pair $b; a$. Now we can perform a standardising permutation, obtaining:

$$\delta'' = \overbrace{(\lambda x.1x)(\underbrace{\ I2\ }_{b})}^{a}(\underbrace{\ I3\ }_{c}) \xrightarrow{a} 1\underbrace{\ I2\ }_{b}(\underbrace{\ I3\ }_{c}) \xrightarrow{b} 12(\underbrace{\ I3\ }_{c}) \xrightarrow{c} 123$$

which is a `s.r.s.`. We can concisely describe the way δ'' is attained as follows:

$$\delta \ = \ b; c; a \stackrel{1}{\Diamond} b; a; c \angle a; b; c \ = \ \delta''$$

Notice that there is another way to standardise δ:

$$\delta \ = \ b; c; a \stackrel{1}{\Diamond} c; b; a \angle c; a; b \ = \ \delta'''$$

resulting in:

$$\delta''' = \overbrace{(\lambda x.1x)(\underbrace{\ I2\ }_{b})}^{a}(\underbrace{\ I3\ }_{c}) \xrightarrow{c} \overbrace{(\lambda x.1x)(\underbrace{\ I2\ }_{b})}^{a}3 \xrightarrow{a} 1(\underbrace{\ I2\ }_{b})3 \xrightarrow{b} 123$$

where $\delta'' \Diamond \delta'''$, as c is disjoint to both a and b. Notice that if a reduction sequence is standard, then any other reduction sequence in its class of square equivalences is standard as well.

We formalise the standardisation process by means of the following definitions.

Definition 2.1.21. *We write $\delta \stackrel{1}{\trianglelefteq} \gamma$ iff $\delta \Diamond \gamma$ or $\delta \angle \gamma$, use $\trianglelefteq$ to denote the reflexive-transitive closure of $\stackrel{1}{\trianglelefteq}$, and say that δ is **more standard** than γ, notation $\delta \lhd \gamma$, iff $\delta \trianglelefteq \delta' \angle \gamma' \trianglelefteq \gamma$. Notice that $\delta \trianglelefteq \gamma$ implies that δ and γ are permutation equivalent.*

By the preorder $\trianglelefteq$, we stratify the $\Diamond$-equivalence classes of reduction sequences by their "standardness degree": if $\delta_1 \Diamond \delta_2$ and $\gamma_1 \Diamond \gamma_2$, then $\delta_1 \trianglelefteq \gamma_1$ iff $\delta_2 \trianglelefteq \gamma_2$. This argument allows to obtain an order[9] $\trianglelefteq / \Diamond$.

Definition 2.1.22. *A reduction sequence is **standard** iff it is contained in a $\Diamond$-equivalence class minimal for $\trianglelefteq / \Diamond$.*

Given this characterisation of `s.r.s.`, in [Mel96] two results relating standardisation with the ARS axioms are stated and proved. Namely:

Theorem 2.1.23. *All ARS enjoying the initial axioms, FD, SO, Linearity and Context-Freeness, verify the following proposition: for any reduction sequence γ, there exists a `s.r.s.` δ such that $\delta \trianglelefteq \gamma$, and therefore δ and γ are permutation equivalent.*

[9] Again, a preorder in the general case, an order in the rewriting systems studied in this thesis. Cfr. [Mel96] Section 4.8.1.

Theorem 2.1.24. *All ARS enjoying the initial axioms, FD, SO, and all the embedding axioms, verify the following proposition: for any reduction sequence γ, there exists a* `s.r.s.` *δ such that $\delta \trianglelefteq \gamma$, and therefore δ and γ are permutation equivalent. Moreover, δ is* unique *modulo $\diamond$, i.e. for any* `s.r.s.` *δ' equivalent with γ, we have $\delta' \diamond \delta$.*

These are among the main results obtained through the ARS model in [Mel96].

2.1.9 A remark on total-order embeddings

We end this section with a remark, which will be important for the study of the *linear substitution calculus* in Chapter 4.

Assume an ARS whose embedding relation is a *total order*, i.e. there are no disjoint steps. Such an embedding can simplify the proofs of the embedding axioms: in the case analysis of the relative embeddings between two steps a and b, if $a \nleq b$, the only possible case left is $b < a$. Particularly, the Stability axiom becomes trivial, since its hypothesis includes the disjointness of two steps.

On the other hand, for such ARS the $\diamond$ equivalence coincides with equality. No square permutations are possible, all permutations are either standardising or anti-standardising. Therefore, the conclusion of Thm. 2.1.24 is stronger: the existence of a *unique* `s.r.s.` equivalent to a given reduction sequence is stated, thus uniqueness is not "modulo $\diamond$".

Hence, an ARS equipped with a total order as its embedding relation leads to a simpler standardisation theory. This fact will be exploited in Chapter 4 to obtain a standardisation result for a rewriting system having a partial-order embedding relation, in two steps. First, a simpler ARS with a total-order is defined. For this ARS, all the axioms required in Thm. 2.1.24 are verified, and thus standardisation is obtained as a corollary of that theorem. Afterwards, this result is used to prove standardisation for the partial-order ARS, by a novel abstract argument. Uniqueness of `s.r.s.` (now modulo square equivalence) is obtained, even though the partial-order ARS does not satisfy all the embedding axioms.

2.2 The proof term model

As described in Section 1.3.2, the intent of the proof term model is to provide a tool to formally denote, or witness, reductions in a given rewriting system. We introduce in this section the main concepts, and some relevant features, of this model, as it is presented for first-order, left-linear term rewriting in [BKdV03], Chapter 8. This is the presentation we extend in Chapter 5 to the realm of infinitary rewriting.

We do not intend to give a complete presentation of the proof term model in this section. The aims of the material we present here are: to give a first glimpse of this generic model of rewriting, including several examples, and to introduce the definitions of some basic notions as given originally for finitary rewriting, to enable the comparison with the infinitary counterparts we introduce in Chapter 5.

In Section 2.2.1 we provide a few basic preliminary definitions, which are essential in order to introduce the proof term model. In Section 2.2.2 we formalise the notion of finitary proof term, providing several examples. In Section 2.2.3 we describe the characterisation of the equivalence of reductions in the proof term model, given in [BKdV03] Section 8.3, which resorts to the notion of *permutation* of contractions. The notion of permutation is also used in the ARS model to describe the equivalence of reductions, cfr. Section 2.1.7. As we pointed out in Section 1.3.2, several characterisations of the equivalence of reductions are proposed in the presentation of the proof term model given in [BKdV03]. We include here only that based in permutations, because it forms the foundation for the characterisation of the equivalence of infinitary reductions we introduce in Section 5.3.

2.2.1 Preliminaries – first-order term rewriting system

Prior to formally introducing proof terms, we must define the notion of first-order term rewriting system. We give here just the definitions we need in Section 2.2.2. For a general presentation of finitary first-order rewriting, cfr. e.g. [BN98], Sections 3.1 and 4.2; and also [BKdV03], Sections 2.1 to 2.3, 2.7 and 2.8. The main concepts of first-order rewriting are defined for infinitary rewriting in this thesis, in Sections 5.1.2 to 5.1.4; cfr. also [BKdV03], Sections 12.1 to 12.3, or [KdV05], Section 2.

Definition 2.2.1 (Signature, function symbol, constant). *A* signature *is a finite set of symbols along with a function from this set to* $\mathbb{N}_{\geqslant 0}$*, called* arity *and noted* ar*. The usual notation is* $\Sigma := \{f_i/n_i\}_{i \in I}$*, where each* f_i *is a symbol and* $n_i = ar(f_i)$*. We follow the custom of writing* $f \in \Sigma$ *as a shorthand notation for* $\exists n . n \in \mathbb{N}_{\geqslant 0} \wedge f/n \in \Sigma$*.*

A constant *is a function symbol* c *such that* $ar(c) = 0$*.*

Definition 2.2.2 (Rewrite rule, term rewriting system). *Assuming a set of variables* `Var` *and given a signature* Σ*, a* rewrite rule *(just* rule *if no confusion arises) over* Σ *is a pair of terms* $\langle l, r \rangle$ *satisfying the following conditions:* $l \notin \mathtt{Var}$*, and each variable occurring in* r *occurs also in* l*. Notation for a rewrite rule:* $l \to r$*, also* $\mu : l \to r$ *if assigning explicit names to rules is desirable. The terms* l *and* r*, respectively, are the* left-hand side *and* right-hand *side,* lhs *and* rhs *for short, of the rule* $l \to r$*.*

A first-order term rewriting system *is a pair* $T = \langle \Sigma, R \rangle$*, where* Σ *is a signature and* R *is a set of rules over* Σ*.*

Definition 2.2.3 (Left-linear term rewriting system). *A term rewriting system is* left-linear *iff for any* l *left-hand side of a rule, and for any* x *variable,* x *occurs in* l *at most once.*

The proof terms we present in the following, as well as the extension to infinitary rewriting we introduce in Chapter 5, apply to left-linear term rewriting systems only.

2.2.2 Proof terms

Proof terms for a given term rewriting system T are terms, in a signature extending that of T. As described in Section 1.3.2, the signature for proof terms includes a *rule symbol* for each rule in T, plus a single binary symbol to denote the *concatenation, or composition*, of reductions. Formally:

Definition 2.2.4 (Signature for proof terms). *Let* $T = \langle \Sigma, R \rangle$ *be a term rewriting system. We define the signature for the proof terms for* T *as follows:* $\Sigma^{PT} := \Sigma \cup \{\mu/n \;/\; \mu : l \to r \in R \wedge |FV(l)| = n\} \cup \{\cdot/2\}$*, where* $FV(l)$ *is the set of variables occurring in* l*. The symbol* $\cdot$*, called the* dot*, denotes the composition, or concatenation, of reductions. It is written infix.*

The set of proof terms for a rewriting system T, along with their source and target terms, can be defined inductively as follows.

Definition 2.2.5 (Proof terms, source, target – [BKdV03], Dfn. 8.2.18)**.** *Let $T = \langle \Sigma, R \rangle$ be a term rewriting system. We say that ψ is a* proof term *for T, and that the terms t and u are the* source *and* target *terms of ψ, iff the conclusion*

$$\psi : t \geq u$$

can be obtained inductively from the following rules:

$$\frac{\psi_1 : s_1 \geq t_1 \quad \dots \quad \psi_n : s_n \geq t_n \quad f/n \in \Sigma}{f(\psi_1, \dots, \psi_n) : f(s_1, \dots, s_n) \geq f(t_1, \dots, t_n)} \ \mathsf{Repl}$$

$$\frac{\psi_1 : s_1 \geq t_1 \quad \dots \quad \psi_n : s_n \geq t_n \quad \rho/n \text{ is a rule symbol} \quad \rho : l \to n}{\rho(\psi_1, \dots, \psi_n) : l[s_1, \dots, s_n] \geq r[t_1, \dots, t_n]} \ \mathsf{Rule}$$

$$\frac{\psi : s \geq t \quad \phi : t \geq u}{\psi \cdot \phi : s \geq u} \ \mathsf{Trans}$$

In the Rule-rule, we employ the following notational convention.

Notation 2.2.6. *In case $\rho : l \to r$ is a rule, $l[s_1, \dots, s_n]$ and $r[s_1, \dots, s_n]$ denote the terms obtained by substituting s_i in l and r, respectively, for the i-th variable of ρ. Here we assume the variables to be ordered in some arbitrary but fixed way depending on ρ. Note that while s_i occurs always exactly once in $l[s_1, \dots, s_n]$, it may occur more than once, or not occur at all, in $r[s_1, \dots, s_n]$, if the corresponding variable appears more than once, or does not appear, in r.*

A convention on terminology follows:

Notation 2.2.7 (Object rewriting system)**.** *When discussing about proof terms for a rewriting system T, we refer to T as the* object *rewriting system. We use the expressions "object signature", "object term" and "object reductions" as well.*

Notice the absence of rules for constants or variables in Dfn. 2.2.5. *Constants* are just symbols in the object signature Σ whose arity is 0; if $a/0 \in \Sigma$, then $a : a \geq a$ can be obtained by just applying the Repl rule. On the other hand, the intent of this definition of proof terms is to model only reductions involving *closed* terms; hence the absence of a rule for *variables*. The restriction to closed terms does not hinder the study of the concepts, particularly the equivalence of reductions, being the aim of the proof term model; to these effects, variable occurrences in a term can be considered as constants. Cfr. [BKdV03], Remark 8.2.21. As a consequence, the inclusion of at least one constant in the signature is required in order to model reductions in a given term rewriting system by using proof terms.

In the following we give several examples of proof terms. Based on these examples, we discuss some features of the proof term model. We use these rules:

$$\mu : f(x) \to g(x) \qquad \nu : g(x) \to k(x) \qquad \rho : h(k(x), y) \to j(y, x)$$

As expected, we can denote the reduction sequence $f(a) \to g(a) \to k(a)$ by the proof term $\mu(a) \cdot \nu(a)$. The corresponding derivation w.r.t. Dfn. 2.2.5 is described in Fig. 2.5.

$$\dfrac{\dfrac{\dfrac{}{a : a \geq a}\,\mathsf{Repl}}{\mu(a) : f(a) \geq g(a)}\,\mathsf{Rule} \qquad \dfrac{\dfrac{}{a : a \geq a}\,\mathsf{Repl}}{\nu(a) : g(a) \geq k(a)}\,\mathsf{Rule}}{\mu(a) \cdot \nu(a) : f(a) \geq k(a)}\,\mathsf{Trans}$$

Figure 2.5: Derivation of a simple proof term

Proof terms can also denote the *simultaneous* contraction of steps. E.g. the term $h(g(a), f(b))$ is the source of two steps, corresponding to the rules μ and ν respectively, which can be contracted simultaneously, yielding $h(k(a), g(b))$ as result. We denote the simultaneous contraction of steps by the decorated arrow $\multimap\!\!\!\to$, so that we write e.g. $h(g(a), f(b)) \multimap\!\!\!\to h(k(a), g(b))$. Cfr. Section 3.1.1 for a discussion of simultaneous contraction and their description in the ARS model. Simultaneous contraction can be also composed with other reductions. Check Fig. 2.6; in this derivation, as well as in those following, some details are omitted.

$$\dfrac{\dfrac{\dfrac{a : a \geq a}{\nu(a) : g(a) \geq k(a)} \qquad \dfrac{b : b \geq b}{\mu(b) : f(b) \geq g(b)}}{h(\nu(a), \mu(b)) \ : \ h(g(a), f(b)) \ \geq \ h(k(a), g(b))}\,\mathsf{Repl} \qquad \rho(a, g(b)) \ : \ h(k(a), g(b)) \ \geq \ j(g(b), a)}{h(\nu(a), \mu(b)) \cdot \rho(a, g(b)) \ : \ h(g(a), f(b)) \ \geq \ j(g(b), a)}\,\mathsf{Trans}$$

$$\dfrac{\ldots \qquad \dfrac{\dfrac{b : b \geq b}{\nu(b) : g(b) \geq k(b)} \qquad a : a \geq a}{j(\nu(b), a) \ : \ j(g(b), a) \ \geq \ j(k(b), a)}\,\mathsf{Repl}}{(h(\nu(a), \mu(b)) \cdot \rho(a, g(b))) \cdot j(\nu(b), a) \ : \ h(g(a), f(b)) \ \geq \ j(k(b), a)}\,\mathsf{Trans}$$

Figure 2.6: Derivation of a proof term involving simultaneous contraction

We can say that the proof term $(h(\nu(a), \mu(b)) \cdot \rho(a, g(b))) \cdot j(\nu(b), a)$ denotes the reduction $h(g(a), f(b)) \multimap\!\!\!\to h(k(a), g(b)) \to j(g(b), a) \to j(k(b), a)$. The following figure depicts the correspondence between (simultaneous) steps and components of the proof term.

$$\begin{array}{ccccccc} h(g(a), f(b)) & \multimap\!\!\!\to & h(k(a), g(b)) & \to & j(g(b), a) & \to & j(k(b), a) \\ & (\ h(\nu(a), \mu(b)) & \cdot & \rho(a, g(b))\ ) & \cdot & j(\nu(b), a) & \end{array}$$

The example given in Fig. 2.7 shows that the steps involved in a simultaneous contraction can be *nested*.

$$\dfrac{\dfrac{a : a \geq a}{\mu(a) : f(a) \geq g(a)}\,\mathsf{Rule} \qquad \dfrac{b : b \geq b}{\mu(b) : f(b) \geq g(b)}\,\mathsf{Rule}}{\rho(\mu(a), \mu(b)) \ : \ h(k(f(a)), f(b)) \ \geq \ j(g(b), g(a))}\,\mathsf{Rule}$$

Figure 2.7: A proof term for simultaneous contraction of nested steps.

Finally, we remark that proof terms allow to denote contractions being performed inside a particular subterm in a term, as the following example shows.

$$\dfrac{\dfrac{\dfrac{\ldots \text{ cfr. Fig. 2.5 } \ldots}{\mu(a) \cdot \nu(a) : f(a) \geq k(a)} \quad b : b \geq b}{h(\mu(a) \cdot \nu(a), b) \;:\; h(f(a), b) \;\geq\; h(k(a), b)}\,\mathsf{Repl} \qquad \rho(a,b) : h(k(a), b) \geq j(b,a)}{h(\mu(a) \cdot \nu(a), b) \cdot \rho(a,b) \;:\; h(f(a), b) \;\geq\; j(b,a)}\,\mathsf{Trans}$$

The preceding examples show that proof terms denote not only reduction sequences, but also different ways in which the contraction of reduction steps can be organised. We use the term *contraction activity* to encompass these different forms of contraction.

We point out that different ways to organise the contraction of the same steps yield different proof terms, implying that the proof term model allows to faithfully denote, and distinguish between, subtly different forms of contraction activity. As an example, let us recall the two steps in the term $h(f(a), g(b))$, cfr. the discussion preceding Fig. 2.6. These steps can be performed sequentially in either order, and their simultaneous contraction is also possible, leading to three different ways to contract these steps. A proof term corresponds to each option, namely: $h(\mu(a), g(b)) \cdot h(g(a), \nu(b))$, $h(f(a), \nu(b)) \cdot h(\mu(a), k(b))$, and $h(\mu(a), \nu(b))$. Note that the source and target terms of all these proof terms coincide. The characterisation of the equivalence of reductions we introduce in the next section yields that these three proof terms are equivalent.

The set of proof terms is a proper subset of the set of terms over the proof term signature. Any term over that signature not including occurrences of the concatenation symbol, i.e. the dot, is a valid proof term, as it can be verified by a simple inductive argument. These proof terms denote the simultaneous contraction of some set of coinitial steps.[10] Particularly, the contraction of a single step is naturally denoted by a proof term with no occurrences of the dot, and with exactly one occurrence of a rule symbol, e.g. $\mu(a)$, $\rho(f(a), g(b))$ or $h(\mu(a), g(b))$. We also remark that all the object terms are valid proof terms, they denote the *trivial* reduction from a term to the same term, not involving any reduction step.

The restrictions shaping the set of valid proof terms are related with the occurrences of the dot, as reflected in the Trans-rule: for $\psi \cdot \phi$ to be a valid proof term, a *coherence condition* applies: the target of ψ and the source of ϕ must coincide. E.g. the term $\mu(a) \cdot \nu(b)$ is not a valid proof term, because the target of $\mu(a)$ and the source of $\nu(b)$, $g(a)$ and $g(b)$ respectively, are different terms.

2.2.3 Equivalence of reductions

Different ways to contract the same steps, regarding sequential versus simultaneous contraction, and/or the sequential ordering in which coinitial steps are performed, yield *equivalent* contraction activities; cfr. the simple example given in Section 1.1.3. As discussed in relation with the ARS model for the particular case of reduction sequences, cfr. Section 2.1.7, the equivalence of reductions can be described in terms of the *permutation*

[10] More precisely, the proof terms without concatenation occurrences denote the simultaneous contraction of coinitial and *mutually orthogonal* sets of steps. We remark that orthogonality of the subjacent term rewriting system is not required.

of contiguous steps: two reduction sequences are considered equivalent iff each of them is the result of a sequence of permutation of steps applied to the other one.

In [BKdV03], Section 8.3, the following equivalence relation on proof terms, which formalises the notion of permutation equivalence for contraction activities, is presented.

Definition 2.2.8 (Permutation equivalence, cfr. [BKdV03] Dfn. 8.3.1)**.** *The* permutation equivalence *relation of proof terms, notation* $\approx$*, is the equivalence and contextual closure of the set of valid instances of the following basic equation schemas:*

$$\begin{array}{lrcl}
(\mathsf{IdLeft}) & src(\psi) \cdot \psi & \sim & \psi \\
(\mathsf{IdRight}) & \psi \cdot tgt(\psi) & \sim & \psi \\
(\mathsf{Assoc}) & \psi \cdot (\phi \cdot \chi) & \sim & (\psi \cdot \phi) \cdot \chi \\
(\mathsf{Struct}) & f(\psi_1, \ldots, \psi_m) \cdot f(\phi_1, \ldots, \phi_m) & \sim & f(\psi_1 \cdot \phi_1, \ldots, \psi_m \cdot \phi_m) \\
(\mathsf{OutIn}) & \mu(\psi_1, \ldots, \psi_m) & \sim & \mu(s_1, \ldots, s_m) \cdot r[\psi_1, \ldots, \psi_m] \\
(\mathsf{InOut}) & \mu(\psi_1, \ldots, \psi_m) & \sim & l[\psi_1, \ldots, \psi_m] \cdot \mu(t_1, \ldots, t_m)
\end{array}$$

where $\mu : l \to r$*,* $s_i = src(\psi_i)$*,* $t_i = tgt(\psi_i)$*, and an instance of an equation is valid iff both the left- and the right-hand sides in that instance are valid proof terms. Cfr. Notation 2.2.6 for the meaning of* $l[\psi_1, \ldots, \psi_n]$ *and* $r[\psi_1, \ldots, \psi_n]$*.*

We remark that this characterisation of permutation equivalence resorts to *equational logic*, applied to proof terms.

The basic equation schemas (Struct), (OutIn) and (InOut) formalise the equivalence of sequential and simultaneous contraction, for parallel steps in the case of (Struct), and for nested steps regarding the latter two. The other equation schemas do not change the organisation of the denoted contraction activity; they are sometimes needed in order to enable the application of some of the other, more significant schemas. Cfr. the *square equivalence* relation in the ARS model, Dfn. 2.1.20.

As a first example, let us consider the rule $\mu : f(x) \to g(x)$, and the proof terms $h(f(a), \mu(b)) \cdot h(\mu(a), g(b))$ and $h(\mu(a), f(b)) \cdot h(g(a), \mu(b))$. These proof terms denote the sequential contraction of the same two, coinitial and parallel, steps, in the two possible orders. Therefore, (the reduction sequences denoted by) these proof terms are equivalent. An abridged permutation equivalence judgement, justifying the equivalence of these proof terms by means of Dfn. 2.2.8, follows.

$$\begin{array}{ll}
\multicolumn{2}{l}{h(f(a), \mu(b)) \cdot h(\mu(a), g(b))} \\
\approx & h(f(a) \cdot \mu(a)\,,\, \mu(b) \cdot g(b)) \\
\approx & h(\mu(a), \mu(b)) \\
\approx & h(\mu(a) \cdot g(a)\,,\, f(b) \cdot \mu(b)) \\
\approx & h(\mu(a), f(b)) \cdot h(g(a), \mu(b))
\end{array}$$

By applying (Struct) and then (IdLeft) and (IdRight), we obtain $h(f(a), \mu(b)) \cdot h(\mu(a), g(b)) \approx h(\mu(a), \mu(b))$, i.e., the equivalence of the sequential and simultaneous contraction of the two involved steps. By means of a similar argument, using the equations in the opposite direction, we obtain that $h(\mu(a), \mu(b)) \approx h(\mu(a), f(b)) \cdot h(g(a), \mu(b))$. In turn, transitivity yields the equivalence between the two sequential proof terms. We can draw some observations from this example:

- The characterisation of the equivalence of reductions given by Dfn. 2.2.8 allows to state not only the equivalence of the two reduction sequences denoted by the original proof terms, but also of both of them with the simultaneous contraction of the involved steps.

- The role of the (IdLeft) and (IdRight) schemas to enable, or complement, the applications of the (Struct) schema can be appreciated.
- In order to prove the equivalence of the two sequential proof terms, starting with one of them, we "pack" the two contracted steps obtaining a proof term denoting their simultaneous contraction, namely $h(\mu(a), \mu(b))$. Subsequently, we "unpack" this simultaneous contraction to obtain the other sequential proof term. Cfr. the description of permutation equivalence in Section 1.3.2.

Let us analyse a second example, involving nested steps. Let us consider the rule $\nu : g(x) \to k(x)$, as well as the μ rule used in the previous example. The proof terms $(f(\mu(a)) \cdot f(\nu(a))) \cdot \mu(k(a))$ and $\mu(f(a)) \cdot (g(\mu(a)) \cdot g(\nu(a)))$ describe the contraction of two nested μ-steps, plus the ν-step created by the contraction of the internal μ-step. The latter proof term can be considered as the result of permuting, in the former one, the external μ-step w.r.t. the two internal steps. The equivalence of these proof terms can be justified by the following permutation equivalence judgement.

$$
\begin{aligned}
(f(\mu(a)) \cdot f(\nu(a))) \cdot \mu(k(a)) & \\
&\approx f(\mu(a)) \cdot (f(\nu(a)) \cdot \mu(k(a))) \\
&\approx f(\mu(a)) \cdot \mu(\nu(a)) && (2.1) \\
&\approx f(\mu(a)) \cdot (\mu(g(a)) \cdot g(\nu(a))) \\
&\approx (f(\mu(a)) \cdot \mu(g(a))) \cdot g(\nu(a)) && (2.2) \\
&\approx \mu(\mu(a)) \cdot g(\nu(a)) && (2.3) \\
&\approx (\mu(f(a)) \cdot g(\mu(a))) \cdot g(\nu(a)) \\
&\approx \mu(f(a)) \cdot (g(\mu(a)) \cdot g(\nu(a)))
\end{aligned}
$$

By applying (Assoc) and then (InOut), we obtain (2.1), which describes the contraction of the inner μ-step followed by the simultaneous contraction of the other two involved steps. The application of (OutIn) and then (Assoc) yields (2.2), in which the external μ-step is permuted with the internal ν-step w.r.t. the original proof term. Subsequently, we apply again (InOut) to obtain (2.3), where the simultaneous contraction of the two μ-steps precedes the ν-step. Finally, by applying again (OutIn) and then (Assoc), we obtain the desired result.

We present an example related with the phenomenon of *erasure*. Consider the rules μ and ν as in the previous example, and $\iota : h(x, y) \to j(y)$, and the reduction sequence

$$h(f(a), g(b)) \xrightarrow{\mu} h(g(a), g(b)) \xrightarrow{\nu} h(g(a), k(b)) \xrightarrow{\iota} j(k(b)) \tag{2.4}$$

where we decorate each arrow with the rule corresponding to each step. The ι step can be permuted with the ν step, resulting in

$$h(f(a), g(b)) \xrightarrow{\mu} h(g(a), g(b)) \xrightarrow{\iota} j(g(b)) \xrightarrow{\nu} j(k(b))$$

In turn, the ι step can be permuted with the μ step also. Applying ι first yields $h(f(a), g(b)) \xrightarrow{\iota} j(g(b))$. The target of this step does not include traces of the source of μ step. The permutation of the ι step w.r.t. the μ step implies the *erasure* of the latter: after the step ι has been performed, there is no step μ to perform. Therefore, the complete result of the permutation is

$$h(f(a), g(b)) \xrightarrow{\iota} j(g(b)) \xrightarrow{\nu} j(k(b)) \tag{2.5}$$

where the μ step has been erased; it is not longer present. A description of the phenomenon of erasure is included in the presentation of the ARS model in this thesis, cfr. Section 2.1.1, particularly Fig. 2.2.

The characterisation of permutation equivalence described in this section, models adequately the erasure of contraction activity. We verify this assertion by formalising the just given example. The reduction sequences (2.4) and (2.5) can be denoted, respectively, by the following proof terms:

$$h(\mu(a), g(b)) \cdot h(g(a), \nu(b)) \cdot \iota(g(a), k(b)) \qquad\qquad \iota(f(a), g(b)) \cdot j(\nu(b))$$

The following abridged derivation proves the equivalence of these proof terms:

$$\begin{aligned} &h(\mu(a), g(b)) \cdot h(g(a), \nu(b)) \cdot \iota(g(a), k(b)) \\ &\qquad\qquad\qquad \approx h(\mu(a), \nu(b)) \cdot \iota(g(a), k(b)) \\ &\qquad\qquad\qquad \approx \iota(\mu(a), \nu(b)) \\ &\qquad\qquad\qquad \approx \iota(f(a), g(b)) \cdot j(\nu(b)) \end{aligned}$$

In this derivation, we first "pack" the μ and ν steps as in the first example given in this section, by applying (Struct), (IdLeft) and (IdRight). In turn, this allows to apply (InOut), obtaining the simultaneous contraction of the three involved steps. Subsequently, we apply (OutIn), yielding the final result.

We note that the instances of (OutIn) corresponding to the ρ rule have the form $\rho(\psi_1, \psi_2) \sim \rho(src(\psi_1), src(\psi_2)) \cdot j(\psi_2)$. If we apply this equation from left to right, as in the last derivation, then the activity denoted by ψ_1, $\mu(a)$ in the example, is erased.

Chapter 3

Normalisation

The subject of this chapter is *normalisation*, that is the computing of normal forms in a given rewriting system, particularly for *non-sequential* systems. The aim is to define *normalising reduction strategies*, described in the introduction, for these systems, and ways to prove that a given strategy is normalising.

The concept of **needed** step is closely related with normalisation. A step in a term t is said to be *needed* if its contraction cannot be avoided when computing a normal form for t. That is, if for any reduction sequence $t \twoheadrightarrow u$ where u is a normal form, either the redex, or at least one of its residuals, is included in the reduction.

A theory of needed redexes is developed in [HL91] for orthogonal first-order term rewriting. For these systems, it is proved that any term not in normal form includes at least one needed redex, and also that systematic reduction of needed redexes is normalising.

Other approaches to normalisation can be subsumed in the concept of needed redexes. Perhaps the first stated result about normalisation, given in [CF58], is that systematic contraction of *leftmost-outermost* redexes is normalising. Consider the term $K\,3\,\Omega$. Contracting the redex Ω yields exactly the same term. Therefore, continuous contraction of (each successive copy of) Ω generates an infinite reduction sequence. On the other hand, the normal form 3 is obtained by a reduction having just two steps if we contract systematically leftmost-outermost redexes. Namely:

$$K\,3\,\Omega \;\rightarrow\; (\lambda y.3)\,\Omega \;\rightarrow\; 3$$

Systematic contraction of leftmost-outermost redexes is also normalising for *left-normal* rewriting systems,[1] as has been proved in [O'D77] and [Klo80] for the first-order and higher-order cases respectively. It is not difficult to show that the leftmost-outermost redex of any term (not in normal form) of any of these rewriting systems is a needed redex.

On the other hand, it is clear that approaches to normalisation based on the concept of needed redex do not apply to rewriting systems which admit terms, not being normal

[1] a rewriting system is *left-normal* iff it is orthogonal and, moreover, for every rewrite rule $t \rightarrow u$, all the occurrences of function and constant symbols in the left-hand side t precede (in the textual rendering of the term) any variable occurrence. E.g. a rewriting system whose unique rule is $f(a,x) \rightarrow b$ is left-normal, while if the rule is $f(x,a) \rightarrow b$ it is not, because the occurrence of a does not precede that of x.

forms, and not including any needed step. A simple example is the *parallel-or* first-order term rewriting system, introduced in Section 2.1.4, cfr. page 34. We recall its rules:

$$\mathtt{or}(x, \mathtt{tt}) \rightarrow \mathtt{tt} \qquad \mathtt{or}(\mathtt{tt}, x) \rightarrow \mathtt{tt}$$

Consider the term $\mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{or}(\mathtt{ff},\mathtt{tt}))$. This term includes two redexes, namely the occurrences of $\mathtt{or}(\mathtt{tt},\mathtt{ff})$ and $\mathtt{or}(\mathtt{ff},\mathtt{tt})$. The reduction sequences to normal form

$$\begin{array}{lclcl} \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{or}(\mathtt{ff},\mathtt{tt})) & \rightarrow & \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{tt}) & \rightarrow & \mathtt{tt} \\ \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{or}(\mathtt{ff},\mathtt{tt})) & \rightarrow & \mathtt{or}(\mathtt{tt},\mathtt{or}(\mathtt{ff},\mathtt{tt})) & \rightarrow & \mathtt{tt} \end{array}$$

show that neither of the two redexes present in the original term is needed: the left (resp. right) redex is not contracted in the first (resp. second) sequence.

This feature in the behavior of the parallel-or rewriting system can be associated with the notions of *sequentiality* and *strong sequentiality* in term rewriting systems. Roughly speaking, a term rewriting system is considered *sequential* iff given an external and "fixed" term structure, say a context C not including redexes, a number i exists such that, for *any term* having the form $C[r_1, \ldots, r_n]$ where all r_i are redexes, the redex r_i is needed. The number i is called an *index* for the context C. In turn, *strongly sequential* term rewriting systems satisfy a stronger condition, which implies that indexes can be effectively computed. Moreover, it is decidable whether a first-order, orthogonal term rewriting system is strongly sequential; cfr. e.g. [HL91]. Different formalisations of the notion of (strongly) sequential term rewriting system have been proposed, cfr. e.g. [HL91, KM89, KM91, SR93]; it is clear that in any case, a rewriting system which admits terms not being normal forms and not having needed redexes, as the parallel-or system, is *non-sequential*.

The example shown for the parallel-or system seems to suggest that no sensible normalising reduction strategy indicating, for a given term, just one step to be contracted, could be built for non-sequential rewriting systems. Indeed, this argument is the motivation to name such systems as *non-sequential*. It should be mentioned, however, that any *almost orthogonal* first-order term rewriting system, such as the parallel-or example, does admit a normalising one-redex strategy, cfr. [Ken89] and [AM96]. There is a price to pay though, namely that such a strategy has to perform lookahead (in the form of cycle detection within terms of a given size).[2] In this work, we are interested in the definition of strategies avoiding such lookahead, as well as the need of keeping the *history* of the previous steps in a reduction sequence. The only information available to a strategy should be the structure and the set of steps of the term it analyses.

Some of the results about normalisation for non-sequential systems found in the literature we are aware of, agree in the convenience of considering **multistep strategies**. Recall from Section 1.1.3 that the "indication" given by a multistep strategy for a given term is a *set* of its redexes, whose contraction is assumed to be performed *simultaneously*.

One of these results is given in [vO99], where normalisation is proved for any *outermost-fair* multistep reduction strategy. This result, which extends a previous one

[2] The existence of such *sequential and normalising* reduction strategies for e.g. the parallel-or rewriting system leads to the following comment included in [Ken89], page 32: "In view of this result, it is not clear that the name 'non-sequential' is appropriate for such systems." Nevertheless, we use the name "non-sequential" with the meaning given above, implying that we consider the parallel-or rewriting system, and also PPC as we will describe shortly, as non-sequential systems.

appeared in [vR97], applies to a large family of higher-order rewriting systems, including non-sequential ones, and described in the generic formalism *HRS*, cfr. [Nip91]. The proof is strongly based on two relations defined on the sets of *positions* of the terms being rewritten. A reduction strategy is said **outermost-fair** iff any outermost step is eventually selected. We observe that the leftmost-outermost reduction strategy for the λ-calculus is *not* outermost fair. Consider the term

$$\Omega\ ((\lambda x.x)3)$$

The leftmost-outermost strategy, given this term, would select the Ω redex, and therefore cycle indefinitely without considering the $(\lambda x.x)3$ redex.

On the other hand, systematic contraction of *necessary* sets of steps is proven normalising in [SR93] for first-order term rewriting. This proof is based on the concepts of residual and nesting between steps. The condition of being a necessary set is a generalisation, to sets of redexes, of the concept of needed step. A set of redexes in a term t is **necessary** if for any reduction $t \twoheadrightarrow u$ where u is a normal form, at least one redex in the set, or one of its residuals, is contracted. Of course, the set of all redexes in a term is indeed necessary; the point is to detect proper subsets being still necessary.

The motivation for the study of normalisation to be presented here is to obtain a systematic way to compute normal forms for the *Pure Pattern Calculus*, PPC in the rest of this chapter.

As mentioned in the introduction, PPC is a pattern calculus allowing any term to be a pattern, and also admitting dynamic pattern formation. The error mechanism of PPC makes it non-sequential. The phenomenon can be already observed in a simpler pattern calculus, allowing only data structures to be patterns. Let us establish that the error mechanism of this calculus consists in yielding the distinguished value $\mathfrak{f}$, which is a normal form. E.g., the contraction of the term $(\lambda \mathtt{a}\, x.x)(\mathtt{b}\,\mathtt{c})$ produces the value $\mathfrak{f}$, that is $(\lambda \mathtt{a}\, x.x)(\mathtt{b}\,\mathtt{c}) \to \mathfrak{f}$, because of the *mismatch* between the pattern $\mathtt{a}\, x$ and the argument $\mathtt{b}\,\mathtt{c}$. Let $\mathtt{p}$ be a ternary data constructor representing a person including her/his name, gender and marital status. For example, $\mathtt{p}\,\mathtt{j}\,\mathtt{m}\,\mathtt{s}$ represents a person name $\mathtt{j}$ (for, say, "Jack"), who is male and single. A function such as $\lambda \mathtt{p}\, x\, \mathtt{m}\,\mathtt{s}.x$ returns the name of any person being male and single, triggering the error mechanism if any other value is given to it. Consider the person $\mathtt{a}$ (for "Alice") being female and divorced, and the following term

$$(\lambda \mathtt{p}\, x\, \mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,(I\mathtt{a})\,(I\mathtt{f})\,(I\mathtt{d}))$$

In this case, the contraction of $(I\mathtt{f})$ yields the argument $\mathtt{p}\,(I\mathtt{a})\,\mathtt{f}\,(I\mathtt{d})$, which is to be matched with the pattern $\mathtt{p}\, x\, \mathtt{m}\,\mathtt{s}$. There is a partial mismatch, between the constants $\mathtt{m}$ and $\mathtt{f}$ in the pattern and argument respectively. In PPC, *any* partial mismatch suffices to trigger the error mechanism. Observe that contraction of $(I\mathtt{d})$ *alone* also suffices to yield a partial mismatch, between the constants $\mathtt{s}$ and $\mathtt{d}$ in this case. This observation leads to the following reduction sequences:

$$\begin{array}{lcll}
(\lambda \mathtt{p}\, x\, \mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,(I\mathtt{a})\,(I\mathtt{f})\,(I\mathtt{d})) & \to & (\lambda \mathtt{p}\, x\, \mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,(I\mathtt{a})\,\mathtt{f}\,(I\mathtt{d})) & \to \quad \mathfrak{f} \\
(\lambda \mathtt{p}\, x\, \mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,(I\mathtt{a})\,(I\mathtt{f})\,(I\mathtt{d})) & \to & (\lambda \mathtt{p}\, x\, \mathtt{m}\,\mathtt{s}.x)(\mathtt{p}\,(I\mathtt{a})\,(I\mathtt{f})\,\mathtt{d}) & \to \quad \mathfrak{f}
\end{array}$$

The first reduction sequence does not contract $I\mathtt{d}$, the second one does not contract $I\mathtt{f}$, and neither contract $I\mathtt{a}$. Therefore the original term does not include any needed redexes. Moreover, notice that the set $\{I\mathtt{d}, I\mathtt{f}\}$ is necessary: the contraction of at least one of them is the requisite to trigger the error mechanism.

Contributions

We present a reduction strategy for PPC, which selects necessary sets of redexes. It is based on the leftmost-outermost strategy for the λ-calculus. Indeed, it coincides with leftmost-outermost if PPC is restricted to the λ-calculus; i.e., if only the translation to PPC of terms in the λ-calculus are considered. Therefore, it is not outermost-fair.

The strategy focuses in the leftmost-outermost *prestep*, i.e. subterm of the form $(\lambda p.s)u$, in a term. If this prestep is a step, then it is the only step selected. Otherwise, as in $(\lambda \mathtt{p}\, x\, \mathtt{m}\, \mathtt{s}.x)\ (\mathtt{p}\,(I\mathtt{a})\,(I\mathtt{f})\,(I\mathtt{d}))$, an analysis yields a set of steps inside the pattern and/or the argument which could provoke the transformation of the outermost prestep into a step. In this way, we obtain a *judicious* strategy for PPC, not being unnecessarily liberal in the sets of redexes it selects.

The other contribution of this chapter is an abstract normalisation proof, described in the ARS model. The proof states that systematic contraction of *necessary* and *non-gripping* sets of redexes is normalising, for ARS verifying a number of axioms. The *non-gripping* condition is, as the name suggests, defined in terms of the gripping relation, and it is the reason for the inclusion of that relation in the present thesis.

The normalisation proof was first developed for PPC. This is the version described in [BKLR12]. In spite of being described for one particular rewriting system, that proof was based in properties about steps, multisteps, residuals, embedding and reduction sequences, which could be described in an abstract way. This fact made it possible to translate the structure of the PPC proof into the abstract setting given by the ARS model. This is the proof we describe in the present chapter.

Due to the features of the defined strategy, and also to the goal of obtaining an abstract result, we use [SR93] as the starting point for the development of our proof.

All the (fundamental, embedding and gripping) axioms of the ARS model described in Section 2.1 are required in our abstract normalisation proof, with the exception of Stability. Moreover, a new axiom, not included in the description of the ARS model in [Mel96], is required as well.

The exclusion of Stability is relevant, since this axiom do not hold for non-sequential systems. On the other hand, the novel axiom allows to complete the analysis of the preservation of embedding in residuals, i.e. the analysis of the relative embeddings of b and c compared to that of b' and c', where $b[\![a]\!]b'$ and $c[\![a]\!]c'$. The new axiom complements the information conveyed by Linearity, Context-Freeness, and Enclave–Embedding.

Plan of the chapter

The material of the first part of this chapter, Sections 3.1 to 3.3, is of an abstract nature, describing the abstract normalisation proof developed in the ARS model. In Section 3.1 we introduce the notions of *multistep* and *multireduction*, formalising them in the ARS model. We also introduce some relations on multisteps and multireductions, and a novel axiom. The material in this section complements the general description of the ARS model given in Section 2.1. In Section 3.2 we define the multistep reduction strategies, and the necessary and non-gripping properties, also in the framework given by the ARS model. Section 3.3 is devoted to the development of the abstract normalisation proof, including all the needed auxiliary results; cfr. Thm. 3.3.14.

Sections 3.4 and 3.5 are focused on PPC. In Section 3.4, we present this calculus, discuss its non-sequential nature, model it in the ARS model, and show that the resulting ARS verifies all the axioms required by our abstract normalisation result. Section 3.5 is devoted to the reduction strategy we propose for PPC: we present this strategy, discuss about its features, and show that the multisteps it selects are necessary and non-gripping. Thus, normalisation of the strategy can be obtained as a consequence of our abstract normalisation result; cfr. Thm. 3.5.26.

3.1 Additional elements of the ARS model

The abstract normalisation proof we developed, requires the introduction of some notions pertaining to the ARS model, which are not included in the description given in Section 2.1. Several of these notions, e.g. those of *multistep* and *multireduction*, are present in [Mel96], while others, as the *free from* and *dominated by* relations, do not appear there. This section is devoted to describe these elements.

3.1.1 Multisteps

Simultaneous contraction of several redexes can be adequately described in the ARS model.

Definition 3.1.1 (Multistep)**.** *A* ***multistep*** *is either an empty set indexed by an object t, notation $\varnothing_t$, or a set of coinitial steps, i.e. a non-empty subset of $\mathcal{RO}(t)$ for a certain object t. We denote such sets by the letters $\mathcal{A}$, $\mathcal{A}'$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, etc.; cfr. Dfn. 2.1.6. We write $t \overset{\mathcal{A}}{\multimap} u$ to indicate that $\mathsf{src}(\mathcal{A}) = t$ and $\mathsf{tgt}(\mathcal{A}) = u$. We use $\mathcal{M}$ to denote the set of multisteps of an ARS.*

The definition of the residuals of a set of coinitial steps, after a step (cfr. Dfn. 2.1.9) or after a reduction sequences (cfr. Dfn. 2.1.13), apply immediately to multisteps. The extension to empty multisteps is straightforward: $\varnothing_{\mathsf{src}(a)}[\![a]\!] = \varnothing_{\mathsf{tgt}(a)}$, $\varnothing_{\mathsf{src}(\delta)}[\![\delta]\!] = \varnothing_{\mathsf{tgt}(\delta)}$. Notice that for any a, b steps, $b[\![a]\!]$ is a multistep; the same happens with $\mathcal{B}[\![a]\!]$ for any multistep $\mathcal{B}$.

A precise definition of the simultaneous contraction of a multistep can be given by resorting to *complete developments*, cfr. Dfn. 2.1.14, for ARS verifying the initial axioms, FD and SO. As noted in Section 2.1.4, the axioms FD and SO imply that all the complete developments of a multistep end in the same target term, and induce the same residual relation. Therefore, some notions defined on steps, like source, target and residuals, can be extended to multisteps.

Definition 3.1.2 (Source, target, residuals for multisteps)**.** *Let $\mathcal{A} \subseteq \mathcal{RO}(t)$ be a multistep, and b a step coinitial with $\mathcal{A}$. If $\mathcal{A} = \varnothing_t$, then we define $\mathsf{src}(\varnothing_t) := \mathsf{tgt}(\varnothing_t) := t$ and $b[\![\mathcal{A}]\!]b'$ iff $b' = b$. Otherwise, we define $\mathsf{src}(\mathcal{A}) := t$, $\mathsf{tgt}(\mathcal{A}) := \mathsf{tgt}(\delta)$, and $b[\![\mathcal{A}]\!]b'$ iff $b[\![\delta]\!]b'$ where δ is an arbitrary complete development of $\mathcal{A}$; cfr. Dfn. 2.1.13.*

Notation 3.1.3. *We use the notations $b[\![\mathcal{A}]\!]$ to denote $\{b' \,/\, b[\![\mathcal{A}]\!]b'\}$, $\mathcal{B}[\![\mathcal{A}]\!]b'$ iff $b[\![\mathcal{A}]\!]b'$ for some $b \in \mathcal{B}$, and $\mathcal{B}[\![\mathcal{A}]\!]$ for $\{b' \,/\, \mathcal{B}[\![\mathcal{A}]\!]b'\}$. Cfr. Notation 2.1.3 and Dfn. 2.1.9.*

Notice that the residual relation is *closed* on multisteps: the residual of a multistep after another is always a multistep. This is not the case for *sets of pairwise disjoint steps* in *higher-order* term rewriting systems. Recall the λ-calculus example used to introduce gripping in Section 1.3.1

$$\overbrace{(\lambda x.\underbrace{Dx}_{b_1})(\overbrace{I3}^{b_2})}^{a} \xrightarrow{a} \underbrace{D(\overbrace{I3}^{b'_2})}_{b'_1}$$

considering the usual nesting on steps. If we define $\mathcal{A} := \{a\}$ and $\mathcal{B} := \{b_1, b_2\}$, we get $\mathcal{B}[\![\mathcal{A}]\!] = \{b'_1, b'_2\}$. The residual of a pairwise disjoint set after another (trivially) pairwise disjoint set, is not in turn disjoint.

For *first-order* rewriting, residuals *are* closed for pairwise disjoint sets of steps. This fact allows a normalisation proof such as the one presented in [SR93], restricted to the first-order case, to be much simpler than the one, more general, to be presented in Section 3.3.

3.1.2 Multireductions

Our normalisation result involves sequences of contractions from a given term to a normal form, formed not by individual steps, but by *multisteps*. This fact requires a precise meaning to be given to such sequences. Fortunately, the concept of *reduction sequence* can be applied, in a natural way, to multisteps as well as to individual steps.

Definition 3.1.4 (Multireduction). *A **multireduction sequence**, or just **multireduction**, is either* $\texttt{nil}_t$*, an* empty multireduction *indexed by the object* t*, or a sequence of multisteps* $\mathcal{A}_1; \ldots; \mathcal{A}_n; \ldots$*. We use* Δ*,* Γ*,* Π*,* Ψ *to denote multireductions and* $\Delta[k]$ *and* $\Delta[i..j]$ *with the same meanings given for reduction sequences. Source and target of multireductions are defined analogously as for reduction sequences; cfr. Dfn. 2.1.10. We will write* $t \overset{\Delta}{\multimap\!\!\!\twoheadrightarrow} u$ *to denote that* $\mathsf{src}(\Delta) = t$ *and* $\mathsf{tgt}(\Delta) = u$*. We use* $\mathcal{MRS}$ *to denote the set of multireductions of an ARS.*

Definition 3.1.5 (Length of a multireduction). *The **length of a multireduction** Δ, notation $|\Delta|$, is the number of multisteps it includes.*

Definition 3.1.6 (Residuals after a multireduction). *The residual relation is extended from multisteps to multireductions, exactly as we have extended it from steps to reduction sequences, cfr. Dfn. 2.1.13. We define* $b[\![\texttt{nil}_t]\!]b$ *for all* $b \in \mathcal{RO}(t)$*, and* $b[\![\mathcal{A};\Delta]\!]b'$ *iff* $b[\![\mathcal{A}]\!]b''$ *and* $b''[\![\Delta]\!]b'$ *for some* b''*. We write* $b[\![\Delta]\!]$ *for* $\{b' \;/\; b[\![\Delta]\!]b'\}$*,* $\mathcal{B}[\![\Delta]\!]b'$ *iff* $b[\![\Delta]\!]b'$ *for some* $b \in \mathcal{B}$*, and* $\mathcal{B}[\![\Delta]\!]$ *for* $\{b' \;/\; \mathcal{B}[\![\Delta]\!]b'\}$*.*

A multireduction is thus a sequence whose elements are, in turn, sets of steps. Notice that the length of a multireduction is is not connected to the sizes of the multisteps which are its elements. An element of a multireduction can be an empty multistep, so that the only corresponding complete development is the empty reduction sequence indexed by its source. We notice that a multireduction consisting of one or more occurrences of $\varnothing_t$, and $\texttt{nil}_t$, are different multireductions. In particular, $|\varnothing_t; \varnothing_t| = 2$ while $|\texttt{nil}_t| = 0$.

Definition 3.1.7 (Trivial multireduction). *We say that a multireduction is **trivial** iff all its elements are empty multisteps. Empty multireductions are trivial.*

Let $\mathcal{A}, \mathcal{B}$ be two multisteps. Recall that $\mathcal{B}\llbracket\mathcal{A}\rrbracket$ is defined as $\{b' \ / \ b\llbracket\delta\rrbracket b'$ where $b \in \mathcal{B}$ and $\delta \Vdash \mathcal{A}\}$, cfr. Notation 3.1.3. Therefore, all the elements in $\mathcal{B}\llbracket\mathcal{A}\rrbracket$ are steps whose source object is $\mathtt{tgt}(\mathcal{B})$, cfr. Dfn. 3.1.1. Hence, $\mathcal{B}\llbracket\mathcal{A}\rrbracket$ is a set of coinitial steps, i.e., a multistep. Put in other words, the residual relation is *closed* on $\mathcal{M}$: the residuals of a multistep after a multistep form, in turn, a multistep. Therefore, we can consider a residual *function* on multisteps, i.e. $\llbracket\cdot\rrbracket : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$. This distinguishing feature of multisteps leads to the following definition.

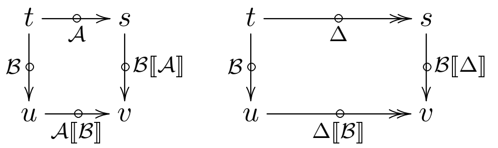
Definition 3.1.8 (Residual of a multireduction after a multistep). *We define the* ***residual of a multireduction after a multistep****, for which we will (ab)use the notation $\llbracket\cdot\rrbracket$, as the following partial function $\mathcal{MRS} \times \mathcal{M} \to \mathcal{MRS}$: if $\mathsf{src}(\mathcal{B}) = t$ then $\mathtt{nil}_t\llbracket\mathcal{B}\rrbracket := \mathtt{nil}_{\mathsf{tgt}(\mathcal{B})}$; if $\mathsf{src}(\mathcal{B}) = \mathsf{src}(\mathcal{A})$ then $(\mathcal{A};\Delta)\llbracket\mathcal{B}\rrbracket := \mathcal{A}\llbracket\mathcal{B}\rrbracket; (\Delta\llbracket\mathcal{B}\llbracket\mathcal{A}\rrbracket\rrbracket)$. Observe that we are defining a* function*, in spite of name "residual" and the notation $\llbracket\cdot\rrbracket$, which correspond in general to* ternary relations*. Notice that $|\Delta\llbracket\mathcal{B}\rrbracket| = |\Delta|$.*

The independence of order of contraction of steps, formalised in Prop. 2.1.16, extends to multisteps [Mel96, Lem. 2.24] and to multireductions. The former is a consequence of Prop. 2.1.16 and the latter then follows by induction on Δ.

Proposition 3.1.9. *Consider an ARS enjoying the group of initial axioms,* FD *and* SO*; cfr. Sections 2.1.3 and 2.1.4.*

1. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{RO}(t)$. The target and residual relation of $\mathcal{A};\mathcal{B}\llbracket\mathcal{A}\rrbracket$ and $\mathcal{B};\mathcal{A}\llbracket\mathcal{B}\rrbracket$ coincide.*
2. *Let Δ be a multireduction, and $\mathcal{B} \subseteq \mathcal{RO}(t)$. The target and residual relation of $\Delta;\mathcal{B}\llbracket\Delta\rrbracket$ and $\mathcal{B};\Delta\llbracket\mathcal{B}\rrbracket$ coincide.*

Graphically:



3.1.3 Some relations on multisteps and multireductions

Two notions related with embedding and involving multisteps are crucial to define the main elements of the abstract normalisation proof. Namely, a step a (resp. a multistep $\mathcal{A}$) is **free from** a multistep $\mathcal{B}$ iff a is not (resp. no step in $\mathcal{A}$ is) embedded by some element of $\mathcal{B}$; while a (resp. $\mathcal{A}$) is **dominated by** $\mathcal{B}$ iff a is (resp. all the steps in $\mathcal{A}$ are) embedded by some element of $\mathcal{B}$. In turn, a *multireduction* Δ is **free from** $\mathcal{B}$, if every element of Δ is free from the respective residual of $\mathcal{B}$. Formally:

Definition 3.1.10 (Free from). *Let $a, \mathcal{A}, \Delta$ be a step, multistep and multireduction, all coinitial with $\mathcal{B}$. We say that*

- *a is* ***free from*** *$\mathcal{B}$, written $a \ddagger \mathcal{B}$, iff $a \not\succeq b$ for all $b \in \mathcal{B}$.*
- *$\mathcal{A}$ is* ***free from*** *$\mathcal{B}$, written $\mathcal{A} \ddagger \mathcal{B}$, iff $a \ddagger \mathcal{B}$ for all $a \in \mathcal{A}$.*
- *Δ is* ***free from*** *$\mathcal{B}$, written $\Delta \ddagger \mathcal{B}$, iff either $\Delta = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$ or $\Delta = \mathcal{A};\Delta'$, $\mathcal{A} \ddagger \mathcal{B}$ and $\Delta' \ddagger \mathcal{B}\llbracket\mathcal{A}\rrbracket$.*

Definition 3.1.11 (Dominated by). *Let $a, \mathcal{A}$ be a step and a multistep, both coinitial with $\mathcal{B}$. We say that*

- *a is **dominated by** $\mathcal{B}$, written $a \rhd \mathcal{B}$, iff $a \notin \mathcal{B}$ and $\exists b \in \mathcal{B} \;/\; a > b$.*
- *$\mathcal{A}$ is **dominated by** $\mathcal{B}$, written $\mathcal{A} \rhd \mathcal{B}$, iff $a \rhd \mathcal{B}$ for all $a \in \mathcal{A}$.*

Notice that being free from and dominated by $\mathcal{B}$ are complementary for a single (coinitial) step a, unless $a \in \mathcal{B}$, i.e. exactly one of $a \in \mathcal{B}$, $a \ddagger \mathcal{B}$ and $a \rhd \mathcal{B}$ holds. This need not be the case for a multistep $\mathcal{A}$: even if $\mathcal{A} \cap \mathcal{B} = \varnothing$, it could well be the case that neither $\mathcal{A} \ddagger \mathcal{B}$ nor $\mathcal{A} \rhd \mathcal{B}$ hold, if some elements of $\mathcal{A}$ are free from $\mathcal{B}$ while others are dominated by it. However, any $\mathcal{A}$ verifying $\mathcal{A} \cap \mathcal{B} = \varnothing$ can be split into a free subset $\mathcal{A}^F$ and a dominated subset $\mathcal{A}^D$ w.r.t. $\mathcal{B}$, i.e. $\mathcal{A} = \mathcal{A}^F \uplus \mathcal{A}^D$, $\mathcal{A}^F \ddagger \mathcal{B}$, and $\mathcal{A}^D \rhd \mathcal{B}$.

Consider the following multireduction in the λ-calculus:

$$\overbrace{(\lambda x.x(\underbrace{I5}_{a}))(\underbrace{I3}_{b})}^{d}\;\overbrace{(I(\underbrace{I4}_{c})}^{e})$$

$$\xrightarrow{\{e\}}\; \overbrace{(\lambda x.x(\underbrace{I5}_{a'}))(\underbrace{I3}_{b'})}^{d'}\;(\underbrace{I4}_{c'}) \;\xrightarrow{\{d',c'\}}\; (\underbrace{I3}_{b''})(\underbrace{I5}_{a''})\;4$$

In this case, we have $\{c,d,e\} \ddagger \{a,b\}$, $\{a,b\} \ddagger \{c,e\}$, $\{a,b,c\} \rhd \{d,e\}$. Moreover, if we take the displayed multireduction, i.e. $\Delta = \{e\};\{d',c'\}$, we have $\Delta \ddagger \{a,b\}$, because $\{e\} \ddagger \{a,b\}$ and $\{d',c'\} \ddagger \{a',b'\}$.

If we define $\mathcal{A} = \{b,c,e\}$ and $\mathcal{B} = \{a,d\}$, we observe that neither $\mathcal{A} \ddagger \mathcal{B}$ nor $\mathcal{A} \rhd \mathcal{B}$ hold. The split of $\mathcal{A}$ w.r.t. $\mathcal{B}$ gives $\mathcal{A}^F = \{c,e\}$ and $\mathcal{A}^D = \{b\}$.

Observe also that being free from a multistep extends to parts of a multireduction, namely:

Lemma 3.1.12. *Assume $\Delta_1;\Delta_2;\Delta_3 \ddagger \mathcal{B}$. Then $\Delta_2 \ddagger \mathcal{B}[\![\Delta_1]\!]$.*

Proof. We proceed by induction on $\langle|\Delta_1|,|\Delta_2|\rangle$. Let Δ be $\Delta_1;\Delta_2;\Delta_3$.

The base case is when $\Delta_1 = \Delta_2 = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$. In this case $\mathcal{B}[\![\Delta_1]\!] = \mathcal{B}$. Then the definition of $\ddagger$ suffices to conclude.

Suppose that $\Delta_1 = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$ and $\Delta_2 = \mathcal{A};\Delta_2'$. In this case, $\Delta = \mathcal{A};\Delta_2';\Delta_3$, so that $\Delta \ddagger \mathcal{B}$ implies $\mathcal{A} \ddagger \mathcal{B}$ and $\Delta_2';\Delta_3 = \mathtt{nil}_{\mathsf{tgt}(\mathcal{A})};\Delta_2';\Delta_3 \ddagger \mathcal{B}[\![\mathcal{A}]\!]$. We observe that $\langle|\mathtt{nil}_{\mathsf{tgt}(\mathcal{A})}|,|\Delta_2'|\rangle < \langle|\Delta_1|,|\Delta_2|\rangle$, therefore we can apply IH, obtaining that $\Delta_2' \ddagger \mathcal{B}[\![\mathcal{A}]\!][\![\mathtt{nil}_{\mathsf{tgt}(\mathcal{A})}]\!] = \mathcal{B}[\![\mathcal{A}]\!]$. Recalling that $\mathcal{A} \ddagger \mathcal{B}$, we get $\Delta_2 \ddagger \mathcal{B} = \mathcal{B}[\![\Delta_1]\!]$.

If $\Delta_1 = \mathcal{A};\Delta_1'$, then $\Delta \ddagger \mathcal{B}$ implies $\mathcal{A} \ddagger \mathcal{B}$ and $\Delta_1';\Delta_2;\Delta_3 \ddagger \mathcal{B}[\![\mathcal{A}]\!]$. Observe $\langle|\Delta_1'|,|\Delta_2|\rangle < \langle|\Delta_1|,|\Delta_2|\rangle$, then IH yields $\Delta_2 \ddagger \mathcal{B}[\![\mathcal{A}]\!][\![\Delta_1']\!] = \mathcal{B}[\![\Delta_1]\!]$. $\square$

Given a multireduction and some coinitial multistep, a further property the abstract normalisation proof is interested in is whether the multistep is at least partially contracted along the multireduction, or if it is otherwise completely ignored. We will say that a multistep is **used** in a multireduction, iff at least one residual of the former is included (i.e. contracted) in the latter. Formally:

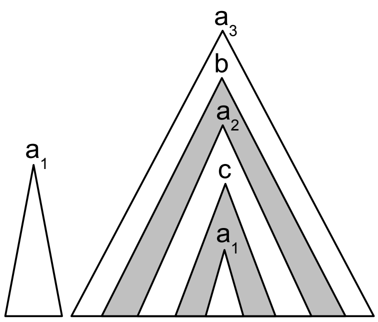
Definition 3.1.13 (Uses). *Let b be a step, $\mathcal{A}$ and $\mathcal{B}$ two multisteps, and Δ a multireduction, such that all of them are coinitial.*

- $\mathcal{A}$ ***uses*** *b iff $b \in \mathcal{A}$;*
- Δ ***uses*** *b iff $\Delta[k] \cap (b[\![\Delta[1..k-1]]\!]) \neq \emptyset$ for at least one k; and*
- *$\mathcal{A}$ (resp. Δ)* ***uses*** *$\mathcal{B}$ iff it uses at least one $b \in \mathcal{B}$.*

3.1.4 A new axiom

The abstract normalisation proof requires the concerned ARS to enjoy a property, related to the preservation of embedding in residuals, which is not implied by the fundamental and normalisation axioms given in Section 2.1, nor included as an additional axiom in [Mel96], since it is not required for any result proven there. We encode this property as a new axiom.

To motivate it, we illustrate an important property that we shall need to prove for our normalisation result. We assume three coinitial redexes a, b, c such that $b < c$ and $a \neq b$, and $c[\![a]\!]c'$ for some c' (cfr. shaded triangles in the figure). We would like to deduce the existence of b' s.t. (i) $b[\![a]\!]b'$ and (ii) $b' < c'$. For that we proceed to consider all possible embedding relations between a, on the one hand, and b and c, on the other (see adjacent figure):



- $a \not< c$. This is represented with the two occurrences of a subscripted with 1. We conclude (i) and (ii) using Linearity and Context-Freeness.
- $a < c$.
  - $b < a$ (hence $b < a < c$). This is represented with the occurrence of a subscripted with 2. We conclude (i) and (ii) using Linearity and Enclave–Embedding.
  - $b \not< a$. This is represented with the occurrence of a subscripted with 3.[3]In this case, the existence of a step b' verifying the required conditions cannot be concluded from the fundamental and normalisation axioms. Hence the need of an additional axiom, to enforce (i) and (ii) in this situation.

The statement of the new axiom follows.

Pivot $a < c$, $b < c$, $b \not\leq a$ and $c[\![a]\!]c' \Rightarrow \exists b' \,/\, b[\![a]\!]b'$ and $b' < c'$.

3.2 Multistep strategies and required properties

The notion of reduction strategy can be described in a simple way in the ARS model. In turn, the notions of step used in a (multi)reduction, and gripping, allow to express the conditions imposed to a reduction strategy in the abstract normalisation proof.

[3]Notice that in the general case, from $a < c$ and $b < c$ one cannot imply $b < a$ or $a \leqslant b$. Such a condition is not implied by the ARS model. In the figure a_3 is nesting b just for graphical simplicity.

Definition 3.2.1 (Reduction strategy). *A **(multistep) reduction strategy** for an ARS $\mathfrak{A}$ is any function $\mathcal{S} : (\mathcal{O} \backslash NF) \to \mathcal{P}(\mathcal{R})$ such that $\mathcal{S}(t) \neq \emptyset$ and $\mathcal{S}(t) \subseteq \mathcal{RO}(t)$ for all t; here NF stands for the set of normal forms of $\mathfrak{A}$; cfr. Dfn. 2.1.8. A **single-step reduction strategy** is a reduction strategy $\mathcal{S}$ s.t. $\mathcal{S}(t)$ is a singleton for every t in the domain of $\mathcal{S}$.*

A multistep reduction strategy determines, for each object, a *multireduction*: if $t \in NF$, then the associated multireduction is $\mathtt{nil}_t$, otherwise it is $\mathcal{S}(t_0); \mathcal{S}(t_1); \ldots; \mathcal{S}(t_n); \ldots$ where $t_0 := t$ and $t_{n+1} := \mathtt{tgt}(\mathcal{S}(t_n))$. The multireduction sequence determined by $\mathcal{S}$ is in fact a *reduction sequence*. These multireductions allow to formally characterise a *normalising reduction strategy*; cfr. Section 1.1.3.

Definition 3.2.2 (Normalising reduction strategy). *A reduction strategy is **normalising** iff for any object t, the determined multireduction ends in a normal form for all normalising objects.*

We formalise *neededness*, and the related notion of *necessary set*, in the ARS model.

Definition 3.2.3 (Needed, necessary). *We say that a step a is **needed** iff for every multireduction $\mathsf{src}(a) \xrightarrow{\Delta} u$ such that u is a normal form, Δ uses a. A multistep $\mathcal{A}$ is **necessary**, iff for every multireduction $\mathsf{src}(\mathcal{A}) \xrightarrow{\Delta} u$ such that u is a normal form, Δ uses $\mathcal{A}$.*

The notion of necessary set generalises that of needed redex; notice that any singleton whose only element is a needed redex is, indeed, a necessary set. As mentioned in the introduction, there is an important difference: while not all terms admit a needed redex, any term admits at least one necessary set, i.e. the set of *all* its redexes.

The other condition to be imposed on reduction strategies is related with *gripping*.

Definition 3.2.4 (Gripping relation on multisteps, non-gripping). *We extend the gripping relation to multisteps as follows:*

- *$\mathcal{B}$ **grips** a, written $a \ll \mathcal{B}$, iff $a \ll b$ for some $b \in \mathcal{B}$.*
- *$\mathcal{B}$ **grips** $\mathcal{A}$, written $\mathcal{A} \ll \mathcal{B}$, iff $a \ll \mathcal{B}$ for at least one $a \in \mathcal{A}$.*

*We declare $\mathcal{B}$ to be **non-gripping** iff for any finite multireduction Ψ such that $\mathsf{src}(\Psi) = \mathsf{src}(\mathcal{B})$, $\mathcal{RO}(\mathsf{tgt}(\Psi)) \not\ll \mathcal{B}[\![\Psi]\!]$. Notice that $\mathcal{B}$ being non-gripping implies that all its residuals are.*

3.3 Necessary normalisation for ARS

We prove in this section that, for any ARS verifying all the fundamental axioms, all the normalisation axioms except for stability, the gripping axioms, and also the new Pivot introduced in Section 3.1.4, the systematic contraction of necessary and non-gripping multisteps is normalising.

The general structure of this proof has been mainly inspired by the normalisation proof for first-order term rewriting systems, given in [SR93]. Assume that $\mathcal{S}$ is a reduction strategy selecting always necessary and non-gripping multisteps. Consider an initial multireduction $t_0 \overset{\Delta_0}{\twoheadrightarrow} u \in \boldsymbol{NF}$, and t_1 the target term of the multistep selected by $\mathcal{S}$ for t_0, i.e. $t_0 \overset{\mathcal{S}(t_0)}{\longrightarrow} t_1$. We construct a multireduction $t_1 \overset{\Delta_1}{\twoheadrightarrow} u$, such that the multireduction Δ_1 is strictly smaller than the original one w.r.t. a convenient well-founded ordering $<$. We have thus transformed the original $t_0 \overset{\Delta_0}{\twoheadrightarrow} u$ in $t_0 \overset{\mathcal{S}(t_0)}{\longrightarrow} t_1 \overset{\Delta_1}{\twoheadrightarrow} u$. Given the well-foundedness of the given ordering on multireductions, an iteration over this procedure allows to conclude that repeated contraction of the multisteps selected by the strategy $\mathcal{S}$ yields the normal form u. This is depicted in Fig. 3.1 where Δ_{k+1} is strictly smaller than Δ_k for all k and Δ_{n+1} is a trivial multireduction. The original multireduction Δ_0 is first transformed into $\mathcal{S}(t_0); \Delta_1$, then successively into $\mathcal{S}(t_0); \ldots; \mathcal{S}(t_k); \Delta_{k+1}$; and finally into $\mathcal{S}(t_0); \ldots; \mathcal{S}(t_n)$.

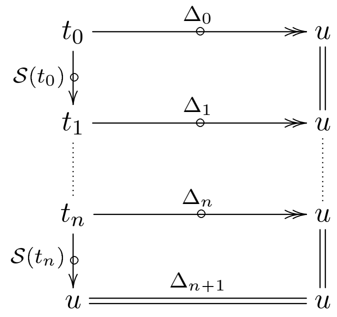


Figure 3.1: Proof idea

Several notions contribute to this proof. We define a **measure** inspired from [SR93, vO99], based on the depths of the multisteps composing a multireduction.

Definition 3.3.1. *Let $\Delta = \Delta[1..n]$ be a multireduction. We define $\chi(\Delta)$ as the n-tuple $\langle \nu(\Delta[n]), \ldots, \nu(\Delta[1]) \rangle$; the lexicographic order is used to compare (measures of) multireductions.*

Notice that the (well-founded) ordering defined allows only to compare multireductions having the same length; the minimal elements are the n-tuples of the form $\langle 0, \ldots, 0 \rangle$ which corresponds exactly to the trivial multireductions. Another observation is that whenever $\chi(\Delta) < \chi(\Gamma)$ then for all multireductions Π, Ψ verifying $\mathsf{tgt}(\Pi) = \mathsf{src}(\Delta)$, $\mathsf{tgt}(\Psi) = \mathsf{src}(\Gamma)$, and $|\Pi| = |\Psi|$, $\chi(\Pi; \Delta) < \chi(\Psi; \Gamma)$ holds. As remarked in [vO99], the measure used in [SR93], based on sizes of multisteps rather than depths, is not well-suited for a higher-order setting.

To build Δ_{k+1}, we observe that the fact that $\mathcal{S}(t_k)$ is a *necessary* set, implies that it is used along Δ_k at least once. Therefore, we can consider the last element of Δ_k, say $\mathcal{A}$, including (some residual of) an element of $\mathcal{S}(t_k)$. We build the diagram shown in Fig. 3.2, where $\Delta_k = \Delta'; \mathcal{A}; \Delta''$, $\mathcal{A} \cap \mathcal{S}(t_k)[\![\Delta']\!] \neq \varnothing$, and Δ'' does not use $\mathcal{S}(t_k)[\![\Delta'; \mathcal{A}]\!]$.

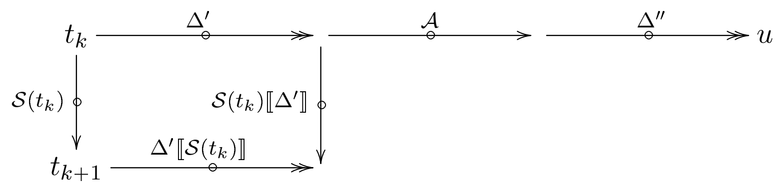


Figure 3.2: Construction of Δ_{k+1}

Let us call $\mathcal{A}_1 = \mathcal{A} \cap \mathcal{S}(t_k)[\![\Delta']\!] \neq \varnothing$, and $\mathcal{A}_2 = (\mathcal{A} \backslash \mathcal{A}_1)[\![\mathcal{A}_1]\!]$. Then we can refine the previous diagram, obtaining Fig. 3.3, where $\mathcal{B} = \mathcal{S}(t_k)[\![\Delta'; \mathcal{A}_1]\!]$. Now $\mathcal{A}_2; \Delta''$ does not use $\mathcal{B}$. Notice that $\mathcal{A}_1 \neq \varnothing$ implies $\nu(\mathcal{A}_2) < \nu(\mathcal{A})$. Observe also that $\mathcal{A}_1 \subseteq \mathcal{S}(t_k)[\![\Delta']\!]$, implying $\mathcal{A}_1[\![\mathcal{S}(t_k)[\![\Delta']\!]]\!] = \varnothing$.

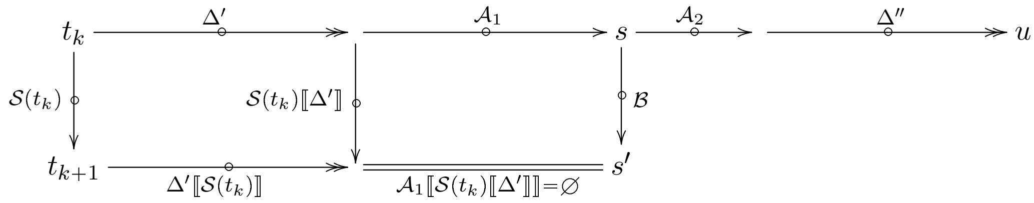


Figure 3.3: Construction of Δ_{k+1}, refined

It suffices to obtain some multireduction Γ' such that $s' \xrightarrow{\Gamma'} u$ and $\chi(\Gamma') \leqslant \chi(\mathcal{A}_2; \Delta'') < \chi(\mathcal{A}; \Delta'')$; taking the elements of a multireduction in *reversed* order in the measure allows to conclude. We obtain the final diagram shown in Fig. 3.4.

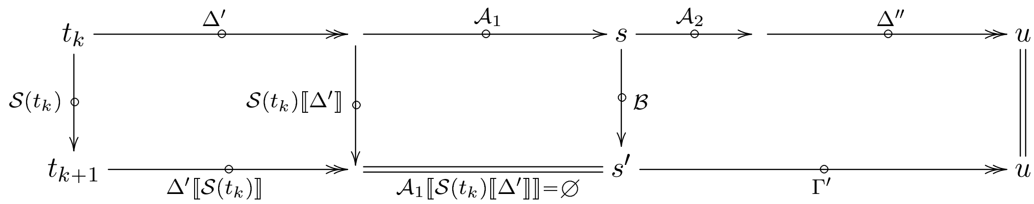


Figure 3.4: Construction of Δ_{k+1}, finished

The construction of Γ' is the most demanding part of the proof. It is based on the following observations:

- Each multistep comprising $\mathcal{A}_2; \Delta''$ can be split in a free and a dominated part w.r.t. $\mathcal{B}$, as remarked in Section 3.1.3.
- Given that $\mathcal{S}(t_k)$ is non-gripping, implying that also $\mathcal{B}$ is non-gripping, the depth of the free part of each multistep can be proven greater or equal than that of its residual after (the corresponding residual of) $\mathcal{B}$. This is the reason for the introduction of gripping.
- Given that $\mathcal{B}$ is not used, and that $u \in \boldsymbol{NF}$ implies $\mathcal{B}[\![\mathcal{A}_2; \Delta'']\!] = \varnothing$, we prove that the dominated part of each multistep can be simply ignored when defining Γ'.

We describe the details in the remainder of this section.

3.3.1 Relevance of gripping

As described in the introduction, one of the effects of having $a \ll b$ is to change the number of residuals of some step c after a. This implies an unwanted change on the depth of certain multisteps. We recall the term involved in the examples on gripping given in the introduction

$$\overbrace{(\lambda x.\underbrace{Dx}_{b})(\underbrace{I3}_{c})}^{a} \xrightarrow{b} \overbrace{(\lambda x.xx)}^{a'}(\underbrace{I3}_{c'})$$

Let us call $\mathcal{A} = \{a, c\}$ and $\mathcal{B} = \{b\}$. Then we have $\mathcal{A} \not\perp \mathcal{B}$ and $\mathcal{A} \ll \mathcal{B}$. Observe that $\nu(\mathcal{A}[\![\mathcal{B}]\!]) = 3 > 2 = \nu(\mathcal{A})$. This increase in the depth of the residual multistep could cause the measure of the multireduction Δ_{k+1}, described in the beginning of this chapter, to be greater than that of Δ_k, thus invalidating the inductive argument of the normalisation proof. The role of gripping in the proof is to guarantee the following property: $\mathcal{A} \not\sqsupset \mathcal{B}$ and $\mathcal{A} \not\ll \mathcal{B}$ imply $\nu(\mathcal{A}[\![\mathcal{B}]\!]) = \nu(\mathcal{A})$. Cfr. Lem. 3.3.4.

Recall that for a set of steps to be non-gripping, any residual of that set of steps must be non-gripping as well, cfr. Dfn. 3.2.4. This requirement in the definition of a non-gripping set is needed to extend Lem. 3.3.4 to multireductions, cfr. Lem. 3.3.5. The proof of these properties follows.

We remark that the requirement about residuals in the definition of a non-gripping set is required also to extend the *postponement* property, again from multisteps to multireductions, cfr. Lem. 3.3.8, and later in Lem. 3.3.10.

Lemma 3.3.2. *Consider $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A} \not\sqsupset \mathcal{B}$, $\mathcal{A} \not\ll \mathcal{B}$, and $d \in \mathcal{A}$. Then $\mathcal{A}[\![d]\!] \not\sqsupset \mathcal{B}[\![d]\!]$ and $\mathcal{A}[\![d]\!] \not\ll \mathcal{B}[\![d]\!]$.*

Proof. If $\mathcal{B} = \varnothing$, then the result holds trivially since also $\mathcal{B}[\![d]\!] = \varnothing$. So assume $b \in \mathcal{B}$. Next, we may assume some $a \in \mathcal{A}$ s.t. $a \neq d$. Otherwise $\mathcal{A}[\![d]\!] = \varnothing$ and the result also holds trivially. For the same reason, $a[\![d]\!]a'$ for some a'. Similarly, we may assume there exists b' s.t. $b[\![d]\!]b'$.

The hypotheses implies the following: $b \nleqslant a$, $b \nleqslant d$, $a \not\ll b$, and $d \not\ll b$.

Observe $b' = a'$ would contradict Ancestor Uniqueness. On the other hand, $b' < a'$ would imply $b < a \vee (d \ll b \wedge d < a)$ by Grip–Instantiation, while $a' \ll b'$ would imply $a \ll b \vee a \ll d \ll b$ by Grip–Density. Therefore, either case would contradict the hypotheses. Thus we conclude. □

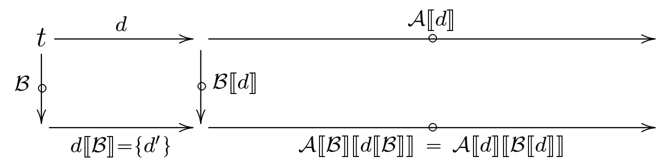
Lemma 3.3.3. *Consider $a, \mathcal{B}$ such that $a \not\sqsupset \mathcal{B}$. Then $a[\![\mathcal{B}]\!]$ is a singleton.*

Proof. By induction on $\nu(\mathcal{B})$. If $\mathcal{B} = \varnothing$, then we conclude by observing that $a[\![\varnothing]\!] = \{a\}$. Otherwise assume some $b \in \mathcal{B}$. Then $a \not\sqsupset \mathcal{B}$ implies $b \nleqslant a$, thus Linearity yields $a[\![b]\!] = \{a'\}$. Let us show that $a' \not\sqsupset \mathcal{B}[\![b]\!]$. Take b'_0 such that $b_0[\![b]\!]b'_0$ for some $b_0 \in \mathcal{B}$. Assume $b'_0 < a'$. Then $b \nleqslant a$ and Context-Freeness imply $b_0 < a$ thus contradicting $a \not\sqsupset \mathcal{B}$. On the other hand, $b'_0 = a'$ would contradict Ancestor Uniqueness. Consequently, $a' \not\sqsupset \mathcal{B}[\![b]\!]$. The IH can then be applied to obtain that $a'[\![\mathcal{B}[\![b]\!]]\!]$ is a singleton. We conclude by observing that $a[\![\mathcal{B}]\!] = a[\![b]\!][\![\mathcal{B}[\![b]\!]]\!] = a'[\![\mathcal{B}[\![b]\!]]\!]$. □

Lemma 3.3.4. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{RO}(t)$ such that $\mathcal{A} \not\sqsupset \mathcal{B}$ and $\mathcal{A} \not\ll \mathcal{B}$. Then $\nu(\mathcal{A}) = \nu(\mathcal{A}[\![\mathcal{B}]\!])$.*

Proof. By induction on $\nu(\mathcal{A})$. If $\mathcal{A} = \varnothing$ then $\mathcal{A}[\![\mathcal{B}]\!] = \varnothing$, thus we immediately conclude. Otherwise we show $\nu(\mathcal{A}) \leqslant \nu(\mathcal{A}[\![\mathcal{B}]\!])$ and $\nu(\mathcal{A}[\![\mathcal{B}]\!]) \leqslant \nu(\mathcal{A})$.

Consider $\delta = d; \delta'$ a complete development of $\mathcal{A}$ verifying $|\delta| = \nu(\mathcal{A})$. Observe that $\delta' \Vdash \mathcal{A}[\![d]\!]$ and moreover $\nu(\mathcal{A}[\![d]\!]) = |\delta'|$, since a development of $\mathcal{A}[\![d]\!]$ longer than δ' would imply $\nu(\mathcal{A}) > |\delta|$. Therefore, $\nu(\mathcal{A}) = \nu(\mathcal{A}[\![d]\!]) + 1$. Lem. 3.3.3 implies $d[\![\mathcal{B}]\!] = \{d'\}$ for some step d'. Furthermore, Prop. 3.1.9:(1) implies $\mathcal{B}; d[\![\mathcal{B}]\!] = d; \mathcal{B}[\![d]\!]$, then $\mathcal{A}[\![\mathcal{B}]\!][\![d[\![\mathcal{B}]\!]]\!] = \mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!]$. Cfr. the following figure:



- We verify $\nu(\mathcal{A}) \leqslant \nu(\mathcal{A}[\![\mathcal{B}]\!])$. Observe that for any complete development γ' of $\mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!]$, $d';\gamma' \Vdash \mathcal{A}[\![\mathcal{B}]\!]$. Moreover, Lem. 3.3.2 implies $\mathcal{A}[\![d]\!] \mathrel{\ddagger} \mathcal{B}[\![d]\!]$ and $\mathcal{A}[\![d]\!] \not\lessdot \mathcal{B}[\![d]\!]$. Then the IH can be applied to $\mathcal{A}[\![d]\!]$ and $\mathcal{B}[\![d]\!]$, yielding $\nu(\mathcal{A}[\![d]\!]) = \nu(\mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!])$. Therefore $\nu(\mathcal{A}) = \nu(\mathcal{A}[\![d]\!]) + 1 = \nu(\mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!]) + 1 \leqslant \nu(\mathcal{A}[\![\mathcal{B}]\!])$.
- We verify $\nu(\mathcal{A}[\![\mathcal{B}]\!]) \leqslant \nu(\mathcal{A})$. Consider $\gamma = d';\gamma'$ a complete development of $\mathcal{A}[\![\mathcal{B}]\!]$ such that $|\gamma| = \nu(\mathcal{A}[\![\mathcal{B}]\!])$. Let $d \in \mathcal{A}$ such that $d[\![\mathcal{B}]\!]d'$. Lem. 3.3.3 implies $d[\![\mathcal{B}]\!] = \{d'\}$, implying $\gamma' \Vdash \mathcal{A}[\![\mathcal{B}]\!][\![d[\![\mathcal{B}]\!]]\!] = \mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!]$. Observe that $\nu(\mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!]) = |\gamma'|$; the contrary would contradict $\nu(\mathcal{A}[\![\mathcal{B}]\!]) = |\gamma|$. Hence $\nu(\mathcal{A}[\![\mathcal{B}]\!]) = \nu(\mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!]) + 1$. An application of the IH similar to that performed earlier yields $\nu(\mathcal{A}[\![d]\!]) = \nu(\mathcal{A}[\![d]\!][\![\mathcal{B}[\![d]\!]]\!])$. Moreover, $\delta' \Vdash \mathcal{A}[\![d]\!]$ implies $d;\delta' \Vdash \mathcal{A}$. Therefore, $\nu(\mathcal{A}[\![\mathcal{B}]\!]) = \nu(\mathcal{A}[\![d]\!]) + 1 \leqslant \nu(\mathcal{A})$.

□

Lem. 3.3.4 can be extended to multireductions.

Lemma 3.3.5. *Let Δ be a multireduction and $\mathcal{B}$ a multistep, such that $\mathsf{src}(\Delta) = \mathsf{src}(\mathcal{B})$, $\mathcal{B}$ is non-gripping and $\Delta \mathrel{\ddagger} \mathcal{B}$. Then $\chi(\Delta) = \chi(\Delta[\![\mathcal{B}]\!])$.*

Proof. By induction on $|\Delta|$. If $\Delta = \mathtt{nil}_{\mathsf{src}(\mathcal{B})}$, then $\Delta[\![\mathcal{B}]\!] = \mathtt{nil}_{\mathsf{tgt}(\mathcal{B})}$, so we conclude immediately. Assume, therefore, $\Delta = \mathcal{A};\Delta'$, so that $\Delta[\![\mathcal{B}]\!] = \mathcal{A}[\![\mathcal{B}]\!];\Delta'[\![\mathcal{B}[\![\mathcal{A}]\!]]\!]$. Observe $\mathcal{A} \mathrel{\ddagger} \mathcal{B}$, $\mathcal{A} \not\lessdot \mathcal{B}$, $\Delta' \mathrel{\ddagger} \mathcal{B}[\![\mathcal{A}]\!]$ and $\mathcal{B}[\![\mathcal{A}]\!]$ is non-gripping. Then Lem. 3.3.4 implies $\nu(\mathcal{A}) = \nu(\mathcal{A}[\![\mathcal{B}]\!])$, and the IH on Δ' yields $\chi(\Delta') = \chi(\Delta'[\![\mathcal{B}[\![\mathcal{A}]\!]]\!])$. Thus we conclude. □

3.3.2 Postponement of dominated multisteps

The next ingredient in the normalisation proof is the ability to *postpone* a dominated multistep after a free multistep or multireduction. The situation is described in Figure 3.5.

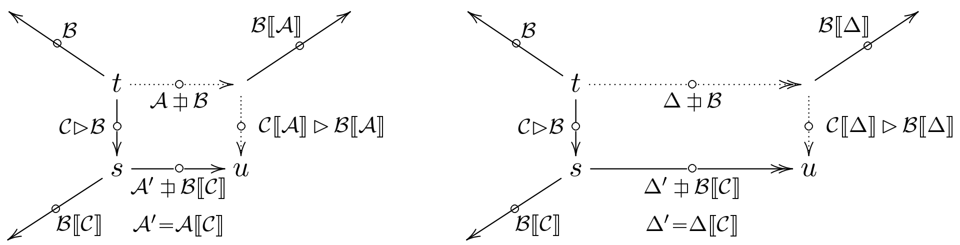


Figure 3.5: Postponement of dominated multisteps: the one step and multiple step case

In the left-hand side diagram, we show the transformation of $\mathcal{C};\mathcal{A}'$ into $\mathcal{A};\mathcal{C}[\![\mathcal{A}]\!]$: a dominated (by $\mathcal{B}$) multistep is *postponed* after a free (from $\mathcal{B}[\![\mathcal{C}]\!]$) one, yielding a multireduction in which the free multistep precedes the dominated one. The right-hand side diagram shows that a dominated multistep can be postponed after a free *multireduction* as well.

We observe that the only role of the added Pivot axiom in the normalisation proof, is to verify that $\mathcal{C}[\![\mathcal{A}]\!] \rhd \mathcal{B}[\![\mathcal{A}]\!]$ in the left-hand side diagram.

The corresponding proofs follow.

Lemma 3.3.6. *Let $\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq \mathcal{RO}(t)$ such that $\mathcal{A} \cap \mathcal{B} = \varnothing$ and $\mathcal{C} \rhd \mathcal{B}$. Then $\mathcal{C}[\![\mathcal{A}]\!] \rhd \mathcal{B}[\![\mathcal{A}]\!]$.*

Proof. We proceed by induction on $\nu(\mathcal{A})$. If $\mathcal{A} = \varnothing$, then $\mathcal{C}[\![\mathcal{A}]\!] = \mathcal{C}$ and $\mathcal{B}[\![\mathcal{A}]\!] = \mathcal{B}$, so that we conclude immediately. Otherwise, consider $a \in \mathcal{A}$ and $c' \in \mathcal{C}[\![a]\!]$ (if $\mathcal{C}[\![a]\!] = \varnothing$, then $\mathcal{C}[\![\mathcal{A}]\!] = \varnothing$ and $\mathcal{C}[\![\mathcal{A}]\!] \rhd \mathcal{B}[\![\mathcal{A}]\!]$ holds trivially). Let $c \in \mathcal{C}$ such that $c' \in c[\![a]\!]$. Note that $a \neq c$ for otherwise $c[\![a]\!] = \varnothing$. We will verify the existence of some $b' \in \mathcal{B}[\![a]\!]$ such that $b' < c'$, so that $\mathcal{C}[\![a]\!] \rhd \mathcal{B}[\![a]\!]$. Let $b \in \mathcal{B}$ be such that $b < c$, as follows from the hypothesis. Observe that $a = b$ or $a = c$ would contradict, respectively, the hypotheses of this lemma or our observation above on the existence of c'. Therefore $a \neq b$ and $a \neq c$. We consider two cases.

1. **Case $a \not< c$.** Then $b < c$ implies $a \not< b$, so that Linearity implies $b[\![a]\!] = \{b'\}$, and then Context-Freeness applies to obtain $b' < c'$.

2. **Case $a < c$.** If $b < a$, i.e. $b < a < c$, then Linearity implies $b[\![a]\!] = \{b'\}$ (since $a \not< b$), and therefore Enclave–Embedding applies to obtain $b' < c'$. Otherwise, we have $a < c$, $b < c$ and $b \not\leq a$, then Pivot applies to obtain $b[\![a]\!]b'$ and $b' < c'$ for some b'.

Hence, we have verified $\mathcal{C}[\![a]\!] \rhd \mathcal{B}[\![a]\!]$. Moreover, Ancestor Uniqueness yields $\mathcal{A}[\![a]\!] \cap \mathcal{B}[\![a]\!] = \varnothing$. Thus we can apply the IH, obtaining $\mathcal{C}[\![a]\!][\![\mathcal{A}[\![a]\!]]\!] \rhd \mathcal{B}[\![a]\!][\![\mathcal{A}[\![a]\!]]\!]$. Thus we conclude. □

Lemma 3.3.7 (Postponement of dominated multisteps – One step case)**.** *Let $\mathcal{B} \subseteq \mathcal{RO}(t)$ and $t \overset{\mathcal{C}}{\circ\!\!\!\longrightarrow} s \overset{\mathcal{A}'}{\circ\!\!\!\longrightarrow} u$, such that $\mathcal{C} \rhd \mathcal{B}$, $\mathcal{A}' \ddagger \mathcal{B}[\![\mathcal{C}]\!]$ and $\mathcal{B}$ is non-gripping. Then there exists $\mathcal{A} \subseteq \mathcal{RO}(t)$ s.t. $\mathcal{A}' = \mathcal{A}[\![\mathcal{C}]\!]$, $\mathcal{A} \ddagger \mathcal{B}$ and $\nu(\mathcal{A}) = \nu(\mathcal{A}')$ (cfr. Fig. 3.5 – left)*

Proof. If $\mathcal{A}' = \varnothing_s$, then taking $\mathcal{A} = \varnothing_t$ suffices to conclude.

If $\mathcal{A}' \neq \varnothing_s$, then we proceed by induction on $\nu(\mathcal{C})$. If $\mathcal{C} = \varnothing$, i.e. $s = t$, then we conclude by taking $\mathcal{A}' := \mathcal{A}$; observe that in this case $\mathcal{B}[\![\mathcal{C}]\!] = \mathcal{B}$.

Consider $c \in \mathcal{C}$ and $t \xrightarrow{c} t_0 \overset{\mathcal{C}[\![c]\!]}{\circ\!\!\!\longrightarrow} s$. Observing that $c \notin \mathcal{B}$ (since $\mathcal{C} \rhd \mathcal{B}$), so that $\{c\} \cap \mathcal{B} = \varnothing$, we can apply Lem. 3.3.6 to obtain $\mathcal{C}[\![c]\!] \rhd \mathcal{B}[\![c]\!]$. Moreover $\mathcal{B}[\![\mathcal{C}]\!] = \mathcal{B}[\![c]\!][\![\mathcal{C}[\![c]\!]]\!]$, and $\mathcal{B}$ non-gripping implies $\mathcal{B}[\![c]\!]$ non-gripping. Therefore, the IH on $\mathcal{C}[\![c]\!]$ yields the existence of some $\mathcal{A}'' \subseteq \mathcal{RO}(t_0)$ such that $\mathcal{A}' = \mathcal{A}''[\![\mathcal{C}[\![c]\!]]\!]$, $\mathcal{A}'' \ddagger \mathcal{B}[\![c]\!]$ and $\nu(\mathcal{A}'') = \nu(\mathcal{A}')$. Hence, to conclude the proof, it suffices to verify the existence of some $\mathcal{A} \subseteq \mathcal{RO}(t)$ verifying (1) $\mathcal{A}'' = \mathcal{A}[\![c]\!]$, (2) $\mathcal{A} \ddagger \mathcal{B}$ and (3) $\nu(\mathcal{A}) = \nu(\mathcal{A}'')$. Observe that $\mathcal{A}' \neq \varnothing_s$ and $\nu(\mathcal{A}'') = \nu(\mathcal{A}')$ imply $\mathcal{A}'' \neq \varnothing_{t_0}$.

1. Let $b_0 \in \mathcal{B}$ such that $b_0 < c$, so that Linearity implies $b_0[\![c]\!] = \{b''_0\}$. Let $a'' \in \mathcal{A}''$. Then a'' being created by c would imply $b''_0 < a''$ by Enclave–Creation, contradicting $\mathcal{A}'' \ddagger \mathcal{B}[\![c]\!]$. Therefore, $a[\![c]\!]a''$ for some a. Let $\mathcal{A} := \{a \mathrel{/} \exists\, a'' \in \mathcal{A}'' .\ a[\![c]\!]a''\}$. Observe that $\mathcal{A}'' \subseteq \mathcal{A}[\![c]\!]$ and let us show that also $\mathcal{A}[\![c]\!] \subseteq \mathcal{A}''$.

   Let $a_0 \in \mathcal{A}[\![c]\!]$, $a \in \mathcal{A}$ such that $a[\![c]\!]a_0$, $a'' \in \mathcal{A}''$ such that $a[\![c]\!]a''$. Observe that $c < a$ would imply $b_0 < c < a$, and then $b''_0 < a''$ by Enclave–Embedding, contradicting $\mathcal{A}'' \ddagger \mathcal{B}[\![c]\!]$. Moreover, $c = a$ would contradict $a[\![c]\!]a''$, cfr. Self Reduction. Therefore $c \not\leq a$, so that Linearity applies yielding that $a[\![c]\!]$ is a singleton, hence $a_0 = a'' \in \mathcal{A}''$. Consequently, $\mathcal{A}[\![c]\!] \subseteq \mathcal{A}''$, and then $\mathcal{A}[\![c]\!] = \mathcal{A}''$.

2. Let $a \in \mathcal{A}$ and $b \in \mathcal{B}$. If b is minimal in $\mathcal{B}$ w.r.t. $<$, then $\mathcal{C} \rhd \mathcal{B}$ implies $b[\![c]\!] = \{b''\}$ by Linearity, since $c \nleqslant b$. Let $a'' \in \mathcal{A}''$ such that $a[\![c]\!]a''$. Observe that we have already verified that $c \nless a$. Then $b < a$ would imply $b'' < a''$ by Context-Freeness, contradicting $\mathcal{A}'' \mathrel{\ddagger} \mathcal{B}[\![c]\!]$. Otherwise, if b is not minimal in $\mathcal{B}$ w.r.t. $<$, then there is some b_0 such that $b_0 < b$ and b_0 is minimal in $\mathcal{B}$ w.r.t. $<$.[4] Therefore, $b_0 \nless a$ implies $b \nless a$. Consequently, $\mathcal{A} \mathrel{\ddagger} \mathcal{B}$.

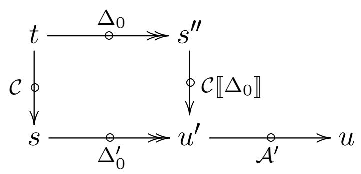
3. Consider $b_0 \in \mathcal{B}$ such that $b_0 < c$. Observe that $a \ll c$ would imply either $a \ll b_0$ or $b_0 \leqslant a$ by Grip–Convexity, contradicting $\mathcal{B}$ being non-gripping and $\mathcal{A} \mathrel{\ddagger} \mathcal{B}$ respectively. Therefore $\mathcal{A} \not\ll c$, and moreover $\mathcal{A} \mathrel{\ddagger} c$ (recall $c \nless a$ for any $a \in \mathcal{A}$). Hence we can apply Lem. 3.3.4 to obtain $\nu(\mathcal{A}) = \nu(\mathcal{A}'')$. Thus we conclude.

□

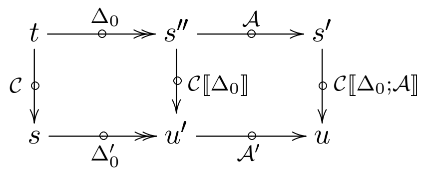
Lemma 3.3.8 (Postponement of dominated multisteps – Multireduction case)**.** *Let $t \xrightarrow{\mathcal{C}} s \overset{\Delta'}{\twoheadrightarrow} u$ and $\mathcal{B} \subseteq \mathcal{RO}(t)$ such that $\mathcal{B}$ is non-gripping, $\mathcal{C} \rhd \mathcal{B}$, and $\Delta' \mathrel{\ddagger} \mathcal{B}[\![\mathcal{C}]\!]$. Then there exists some multireduction Δ verifying $\Delta' = \Delta[\![\mathcal{C}]\!]$, so that $t \overset{\Delta}{\twoheadrightarrow} s' \xrightarrow{\mathcal{C}[\![\Delta]\!]} u$ for some object s' (cfr. Prop. 3.1.9:(2)), and moreover $\Delta \mathrel{\ddagger} \mathcal{B}$, $\mathcal{C}[\![\Delta]\!] \rhd \mathcal{B}[\![\Delta]\!]$, and $\chi(\Delta) = \chi(\Delta')$. The effect is that a multistep dominated by $\mathcal{B}$ is* postponed *after a multireduction free from the same $\mathcal{B}$, without affecting neither the free-from and domination relations w.r.t. (the corresponding residual of) $\mathcal{B}$, nor the measure of the "free" multireduction (cfr. Fig. 3.5 – right).*

Proof. We proceed by induction on $|\Delta'|$. If $\Delta' = \mathtt{nil}_s$, i.e. $u = s$, then it suffices to take $\Delta := \mathtt{nil}_t$, so that $s' = t$.

Assume $\Delta' = \Delta'_0; \mathcal{A}'$, so that $t \xrightarrow{\mathcal{C}} s \overset{\Delta'_0}{\twoheadrightarrow} u' \xrightarrow{\mathcal{A}'} u$. Observe that Lem. 3.1.12 implies $\Delta'_0 \mathrel{\ddagger} \mathcal{B}[\![\mathcal{C}]\!]$. Then we can apply the IH on Δ'_0 obtaining that $\Delta'_0 = \Delta_0[\![\mathcal{C}]\!]$ for some multireduction Δ_0, so that $t \overset{\Delta_0}{\twoheadrightarrow} s'' \xrightarrow{\mathcal{C}[\![\Delta_0]\!]} u'$ for some object s'', and moreover $\Delta_0 \mathrel{\ddagger} \mathcal{B}$, $\mathcal{C}[\![\Delta_0]\!] \rhd \mathcal{B}[\![\Delta_0]\!]$, and $\chi(\Delta_0) = \chi(\Delta'_0)$. We can build the following diagram.



On the other hand, $\Delta' \mathrel{\ddagger} \mathcal{B}[\![\mathcal{C}]\!]$ implies $\mathcal{A}' \mathrel{\ddagger} \mathcal{B}[\![\mathcal{C}; \Delta'_0]\!]$ (cfr. again Lem. 3.1.12), therefore Prop. 3.1.9:(2) yields $\mathcal{A}' \mathrel{\ddagger} \mathcal{B}[\![\Delta_0; \mathcal{C}[\![\Delta_0]\!]]\!] = \mathcal{B}[\![\Delta_0]\!][\![\mathcal{C}[\![\Delta_0]\!]]\!]$. Moreover, $\mathcal{B}$ non-gripping implies $\mathcal{B}[\![\Delta_0]\!]$ non-gripping. Hence we can apply Lem. 3.3.7 to $s'' \overset{\mathcal{C}[\![\Delta_0]\!]}{\twoheadrightarrow} u'$, obtaining that $\mathcal{A}' = \mathcal{A}[\![\mathcal{C}[\![\Delta_0]\!]]\!]$ for some $\mathcal{A} \subseteq \mathcal{RO}(s'')$ verifying $\mathcal{A} \mathrel{\ddagger} \mathcal{B}[\![\Delta_0]\!]$ and $\nu(\mathcal{A}) = \nu(\mathcal{A}')$. Consequently, we can complete the previous diagram as follows.



[4]The fact that $<$ is a well-founded order, cfr. Dfn. 2.1.1, implies the existence, for any $b \in \mathcal{B}$, of a *minimal* $b_0 \in \mathcal{B}$ w.r.t. $<$ such that $b_0 < b$, unless b is itself $<$-minimal in $\mathcal{B}$.

We define $\Delta := \Delta_0; \mathcal{A}$. Given $\mathcal{C}[\![\Delta_0]\!] \rhd \mathcal{B}[\![\Delta_0]\!]$ and $\mathcal{A} \ddagger \mathcal{B}[\![\Delta_0]\!]$, so that $\mathcal{A} \cap \mathcal{B}[\![\Delta_0]\!] = \varnothing$, Lem. 3.3.6 implies that $\mathcal{C}[\![\Delta_0]\!][\![\mathcal{A}]\!] \rhd \mathcal{B}[\![\Delta_0]\!][\![\mathcal{A}]\!]$, that is, $\mathcal{C}[\![\Delta]\!] \rhd \mathcal{B}[\![\Delta]\!]$. Moreover, given $\Delta_0 \ddagger \mathcal{B}$ and $\mathcal{A} \ddagger \mathcal{B}[\![\Delta_0]\!]$, a simple induction on $|\Delta_0|$ yields $\Delta \ddagger \mathcal{B}$. Finally, $\chi(\Delta) = \chi(\Delta')$ is immediate. Thus we conclude. □

3.3.3 Main results

The postponement result is used to show that, whenever $t \overset{\Delta}{\twoheadrightarrow} u$, $\mathcal{B} \subseteq \mathcal{RO}(t)$ is non-gripping and not used in Δ, and $\mathcal{B}[\![\Delta]\!] = \varnothing$, all activity dominated by (the successive residuals of) $\mathcal{B}$ is irrelevant, i.e. it can be omitted without compromising the target object u, and moreover without increasing the measure. Therefore, the dominated part of each multistep in $\mathcal{A}_2; \Delta''$ can be just discarded in the construction of Δ_{k+1}, cfr. Figure 3.4 on page 62.

Lemma 3.3.9. *Let $t \xrightarrow{\mathcal{C}} s \overset{\Delta'}{\twoheadrightarrow} u$ and $\mathcal{B} \subseteq \mathcal{RO}(t)$, such that $\mathcal{B}$ is non-gripping, $\mathcal{C} \rhd \mathcal{B}$, $\Delta' \ddagger \mathcal{B}[\![\mathcal{C}]\!]$, and $\mathcal{B}[\![\mathcal{C}; \Delta']\!] = \varnothing$. Then there is a multireduction Δ such that $\Delta' = \Delta[\![\mathcal{C}]\!]$, $t \overset{\Delta}{\twoheadrightarrow} u$, $\Delta \ddagger \mathcal{B}$, $\mathcal{B}[\![\Delta]\!] = \varnothing$ and $\chi(\Delta) = \chi(\Delta')$.*

Proof. Lem. 3.3.8 implies the existence of Δ such that $\Delta' = \Delta[\![\mathcal{C}]\!]$, $t \overset{\Delta}{\twoheadrightarrow} s' \xrightarrow{\mathcal{C}[\![\Delta]\!]} u$, $\Delta \ddagger \mathcal{B}$, $\mathcal{C}[\![\Delta]\!] \rhd \mathcal{B}[\![\Delta]\!]$, and $\chi(\Delta) = \chi(\Delta')$. Then $\mathcal{B}[\![\Delta]\!][\![\mathcal{C}[\![\Delta]\!]]\!] = \mathcal{B}[\![\Delta; \mathcal{C}[\![\Delta]\!]]\!] = \mathcal{B}[\![\mathcal{C}; \Delta']\!] = \varnothing$; cfr. Prop. 3.1.9:(2).

Assume for contradiction the existence of some $b \in \mathcal{B}[\![\Delta]\!]$, and moreover that (wlog) b is minimal in $\mathcal{B}[\![\Delta]\!]$ w.r.t. $<$. Then $\mathcal{C}[\![\Delta]\!] \rhd \mathcal{B}[\![\Delta]\!]$ implies $b \ddagger \mathcal{C}[\![\Delta]\!]$, so that Lem. 3.3.3 yields $b[\![\mathcal{C}[\![\Delta]\!]]\!] = \{b'\}$, contradicting $(\mathcal{B}[\![\Delta]\!])[\![\mathcal{C}[\![\Delta]\!]]\!] = \varnothing$. Therefore $\mathcal{B}[\![\Delta]\!] = \varnothing$.

In turn, the existence of some $c \in \mathcal{C}[\![\Delta]\!]$ would imply that of some $b \in \mathcal{B}[\![\Delta]\!]$ such that $b < c$, contradicting $\mathcal{B}[\![\Delta]\!] = \varnothing$. Therefore $\mathcal{C}[\![\Delta]\!] = \varnothing$, implying $u = s'$ so that $t \overset{\Delta}{\twoheadrightarrow} u$. Thus we conclude. □

Lemma 3.3.10. *Let $t \overset{\Delta}{\twoheadrightarrow} u$ and $\mathcal{B} \subseteq \mathcal{RO}(t)$, such that $\mathcal{B}$ is non-gripping, Δ does not use $\mathcal{B}$, and $\mathcal{B}[\![\Delta]\!] = \varnothing$. Then there exists a multireduction Γ such that $t \overset{\Gamma}{\twoheadrightarrow} u$, $\Gamma \ddagger \mathcal{B}$, $\mathcal{B}[\![\Gamma]\!] = \varnothing$ and $\chi(\Gamma) \leqslant \chi(\Delta)$.*

Proof. We proceed by induction on $|\Delta|$. If $\Delta = \mathtt{nil}_t$, then it suffices to take $\Gamma := \Delta$.

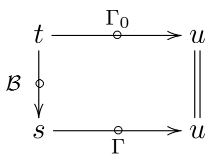
Assume $\Delta = \mathcal{A}; \Delta_0$, so that $t \xrightarrow{\mathcal{A}} s \overset{\Delta_0}{\twoheadrightarrow} u$ for some object s. Observe $\mathcal{B}[\![\mathcal{A}]\!]$ is non-gripping. Then we can apply the IH on $s \overset{\Delta_0}{\twoheadrightarrow} u$, thus obtaining $s \overset{\Gamma'_0}{\twoheadrightarrow} u$ for some Γ'_0 verifying $\Gamma'_0 \ddagger \mathcal{B}[\![\mathcal{A}]\!]$, $\mathcal{B}[\![\mathcal{A}]\!][\![\Gamma'_0]\!] = \varnothing$ and $\chi(\Gamma'_0) \leqslant \chi(\Delta_0)$.

We define $\mathcal{A}^F := \{a \in \mathcal{A} \,/\, a \ddagger \mathcal{B}\}$ and $\mathcal{A}^D := (\mathcal{A} \backslash \mathcal{A}^F)[\![\mathcal{A}^F]\!]$, so that $t \xrightarrow{\mathcal{A}^F} t' \xrightarrow{\mathcal{A}^D} s \overset{\Gamma'_0}{\twoheadrightarrow} u$ for some object t'. As mentioned in Section 3.1.3, it is easy to check $\mathcal{A}^F \ddagger \mathcal{B}$ and $(\mathcal{A} \backslash \mathcal{A}^F) \rhd \mathcal{B}$; recall $\mathcal{A}$ does not use $\mathcal{B}$, so that $\mathcal{A} \cap \mathcal{B} = \varnothing$. As moreover $\mathcal{A}^F \cap \mathcal{B} = \varnothing$, then Lem. 3.3.6 yields $\mathcal{A}^D \rhd \mathcal{B}[\![\mathcal{A}^F]\!]$. Observe that $\mathcal{B}$ non-gripping implies $\mathcal{B}[\![\mathcal{A}^F]\!]$ non-gripping, $\Gamma'_0 \ddagger \mathcal{B}[\![\mathcal{A}]\!] = \mathcal{B}[\![\mathcal{A}^F]\!][\![\mathcal{A}^D]\!]$, and $\mathcal{B}[\![\mathcal{A}^F]\!][\![\mathcal{A}^D; \Gamma'_0]\!] = \mathcal{B}[\![\mathcal{A}]\!][\![\Gamma'_0]\!] = \varnothing$; cfr. Prop. 2.1.16. Therefore Lem. 3.3.9 applies to $t' \xrightarrow{\mathcal{A}^D} s \overset{\Gamma'_0}{\twoheadrightarrow} u$, implying the existence of some Γ_0 verifying $t' \overset{\Gamma_0}{\twoheadrightarrow} u$, $\Gamma_0 \ddagger \mathcal{B}[\![\mathcal{A}^F]\!]$, $\mathcal{B}[\![\mathcal{A}^F]\!][\![\Gamma_0]\!] = \varnothing$ and $\chi(\Gamma_0) = \chi(\Gamma'_0) \leqslant \chi(\Delta_0)$. Hence we conclude by taking $\Gamma := \mathcal{A}^F; \Gamma_0$ since $\mathcal{A}^F \subseteq \mathcal{A}$ implies in particular that $\nu(\mathcal{A}^F) \leqslant \nu(\mathcal{A})$. □

We now conclude the normalisation proof, following the main lines given at the beginning of this section.

Proposition 3.3.11. *Let $t \overset{\Delta}{\twoheadrightarrow} u$ and $\mathcal{B} \subseteq \mathcal{RO}(t)$ s.t. $\mathcal{B}$ is non-gripping, Δ does not use $\mathcal{B}$, $\mathcal{B}[\![\Delta]\!] = \varnothing$ and $t \xrightarrow{\mathcal{B}} s$. Then there exists a multireduction Γ s.t. $s \overset{\Gamma}{\twoheadrightarrow} u$ and $\chi(\Gamma) \leqslant \chi(\Delta)$.*

Proof. Lem. 3.3.10 implies the existence of some Γ_0 such that $t \overset{\Gamma_0}{\twoheadrightarrow} u$, $\Gamma_0 \not\ddagger \mathcal{B}$, $\mathcal{B}[\![\Gamma_0]\!] = \varnothing$ and $\chi(\Gamma_0) \leqslant \chi(\Delta)$. We define $\Gamma := \Gamma_0[\![\mathcal{B}]\!]$. Then we can build the following diagram; cfr. Prop. 3.1.9(2).



Lem. 3.3.5 implies $\chi(\Gamma) = \chi(\Gamma_0) \leqslant \chi(\Delta)$. Thus we conclude. □

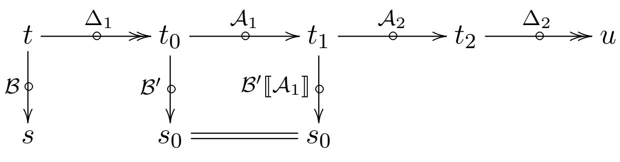
Lemma 3.3.12. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{RO}(t)$ such that $\mathcal{A} \neq \varnothing$ and $\mathcal{A} \cap \mathcal{B} = \varnothing$. Then $\nu(\mathcal{B}[\![\mathcal{A}]\!]) < \nu(\mathcal{A} \cup \mathcal{B})$.*

Proof. Let δ be a complete development of $\mathcal{B}[\![\mathcal{A}]\!]$ such that $|\delta| = \nu(\mathcal{B}[\![\mathcal{A}]\!])$, and γ a complete development of $\mathcal{A}$. Observe that $\gamma; \delta$ is a complete development of $\mathcal{A} \cup \mathcal{B}$, and $|\gamma| > 0$ since $\mathcal{A} \neq \varnothing$. Then $\nu(\mathcal{B}[\![\mathcal{A}]\!]) = |\delta| < |\gamma; \delta| \leqslant \nu(\mathcal{A} \cup \mathcal{B})$. Thus we conclude. □

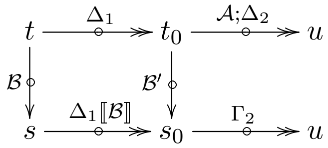
Proposition 3.3.13. *Let $t \overset{\Delta}{\twoheadrightarrow} u$ and $\mathcal{B} \subseteq \mathcal{RO}(t)$, s.t. $\mathcal{B}$ is non-gripping, Δ uses $\mathcal{B}$, $\mathcal{B}[\![\Delta]\!] = \varnothing$ and $t \xrightarrow{\mathcal{B}} s$. Then there exists a multireduction Γ such that $s \overset{\Gamma}{\twoheadrightarrow} u$ and $\chi(\Gamma) < \chi(\Delta)$.*

Proof. The hypothesis indicates Δ uses $\mathcal{B}$, therefore the "last" multistep of Δ which uses the corresponding residual of $\mathcal{B}$ can be determined, i.e. Δ can be written as $\Delta_1; \mathcal{A}; \Delta_2$, such that $\mathcal{A}$ uses $\mathcal{B}[\![\Delta_1]\!]$ (i.e. $\mathcal{A} \cap \mathcal{B}[\![\Delta_1]\!] \neq \varnothing$) and Δ_2 does not use $\mathcal{B}[\![\Delta_1; \mathcal{A}]\!]$. Observe $|\Delta| = |\Delta_1| + |\Delta_2| + 1$.

Let $\mathcal{B}' := \mathcal{B}[\![\Delta_1]\!]$, $\mathcal{A}_1 := \mathcal{A} \cap \mathcal{B}'$, and $\mathcal{A}_2 := (\mathcal{A} \backslash \mathcal{A}_1)[\![\mathcal{A}_1]\!]$. Observe that $\mathcal{A}_1 \neq \varnothing$, so that Lem. 3.3.12 implies $\nu(\mathcal{A}_2) < \nu(\mathcal{A})$. Therefore $\chi(\mathcal{A}_2; \Delta_2) < \chi(\mathcal{A}; \Delta_2)$. Moreover $\mathcal{A}_1[\![\mathcal{B}']\!] = \varnothing$. We can build the following diagram



Suppose $\mathcal{A}_2$ uses $\mathcal{B}'[\![\mathcal{A}_1]\!]$. Notice that the existence of some $b' \in \mathcal{A}_2 \cap \mathcal{B}'[\![\mathcal{A}_1]\!]$ would in turn imply the existence of some $b_1 \in \mathcal{B}'$ s.t. $b_1[\![\mathcal{A}_1]\!]b'$ and also the existence of some $b_2 \in \mathcal{A} \backslash \mathcal{A}_1$ s.t. $b_2[\![\mathcal{A}_1]\!]b'$. Consider an arbitrary $\delta \Vdash \mathcal{A}_1$. Then a simple induction on $|\delta|$, based on Ancestor Uniqueness, yields $b_1 = b_2$. Therefore $b_1 = b_2 \in \mathcal{B}' \cap (\mathcal{A} \backslash \mathcal{A}_1)$. But then, by definition of $\mathcal{A}_1$, $b_1 = b_2 \in \mathcal{A}_1$, which is absurd. Therefore $\mathcal{A}_2$ does not use $\mathcal{B}'[\![\mathcal{A}_1]\!]$ and hence, since Δ_2 does not use $\mathcal{B}[\![\Delta_1; \mathcal{A}]\!]$, $\mathcal{A}_2; \Delta_2$ does not use $\mathcal{B}'[\![\mathcal{A}_1]\!]$. Moreover, $\mathcal{B}$ non-gripping implies $\mathcal{B}'[\![\mathcal{A}_1]\!]$ non-gripping. Hence Prop.. 3.3.11 yields the existence of some Γ_2 verifying $s_0 \xrightarrow{\Gamma_2} u$ and $\chi(\Gamma_2) \leqslant \chi(\mathcal{A}_2; \Delta_2) < \chi(\mathcal{A}; \Delta_2)$. Observe $|\Gamma_2| = |\Delta_2| + 1$.



Thus if we define $\Gamma := \Delta_1[\![\mathcal{B}]\!]; \Gamma_2$, then $|\Gamma| = |\Delta_1| + |\Delta_2| + 1 = |\Delta|$, and $\chi(\Gamma_2) < \chi(\mathcal{A}; \Delta_2)$ implies $\chi(\Gamma) < \chi(\Delta)$ independently of the relative measures of $\Delta_1[\![\mathcal{B}]\!]$ and Δ_1, since the multisteps of a multireduction are considered in *reversed order* when building measures. Thus we conclude. □

Returning to the general proof structure described at the beginning of the section, Prop. 3.3.13 shows the existence of an adequate Δ_{k+1}; consider t_k, t_{k+1}, $\mathcal{S}(t_k)$ and Δ_k as t, s, $\mathcal{B}$ and Δ respectively in the statement of that proposition.

Theorem 3.3.14. *Let $\mathfrak{A} = \langle \mathcal{O}, \mathcal{R}, \mathsf{src}, \mathsf{tgt}, [\![\cdot]\!], <, \ll \rangle$ be an ARS enjoying all the fundamental axioms, all the embedding axioms except for* Stability*, all the gripping axioms, and* Pivot*. Repeated contraction of necessary and non-gripping multisteps on $\mathfrak{A}$ normalises.*

Proof. Let $t_0 \in \mathcal{O}$ be a normalising object in $\mathfrak{A}$. Then there exists some multireduction Δ_0 such that $t_0 \xrightarrow{\Delta_0} u$ where u is a normal form. We proceed by induction on $\chi(\Delta_0)$, i.e. using the well-founded ordering defined in the beginning of this section. If $\chi(\Delta_0)$ is minimal, i.e. either $\Delta_0 = \mathtt{nil}_{t_0}$ or $\Delta_0 = \langle \varnothing_{t_0}, \ldots, \varnothing_{t_0} \rangle$, then t_0 is a normal form, and therefore there is nothing to prove. Otherwise, let $\mathcal{B}$ be a necessary and non-gripping multistep such that $t_0 \xrightarrow{\mathcal{B}} t_1$. Then Δ_0 uses $\mathcal{B}$, and u being a normal form implies $\mathcal{B}[\![\Delta_0]\!] = \varnothing$. Therefore Prop. 3.3.13 implies the existence of a multireduction Δ_1 such that $t_1 \xrightarrow{\Delta_1} u$ and $\chi(\Delta_1) < \chi(\Delta_0)$. The IH on Δ_1 suffices to conclude. □

3.4 The Pure Pattern Calculus

As mentioned in the introduction, PPC is a pattern calculus allowing arbitrary terms to be used as patterns, and supporting novel forms of polymorphism; cfr. [JK09] where several examples are included. In this section, we present this calculus. We first present a brief overview of PPC following [JK09]. Then, we show that PPC fits the ARS framework, including all the axioms required by the abstract normalisation result.

The next section is devoted to the presentation of a normalising reduction strategy for PPC.

3.4.1 Overview of PPC

Consider a countable set of **symbols** $f, g, \ldots, x, y, z$. Sets of symbols are denoted by meta-variables θ, ϕ, ... The syntax of PPC is summarised by the following grammar:

Terms	(***T***)	t	::=	$x \mid \widehat{x} \mid tt \mid \lambda_\theta\ t.t$
Data-Structures	(***DS***)	D	::=	$\widehat{x} \mid Dt$
Abstractions	(***ABS***)	A	::=	$\lambda_\theta\ t.t$
Matchable-forms	(***MF***)	F	::=	$D \mid A$

The term x is called a **variable**, $\widehat{x}$ a **matchable**, tu an **application** (t is the **function** and u the **argument**) and $\lambda_\theta\ p.u$ an **abstraction** (θ is the set of **binding symbols**, p is the **pattern** and u is the **body**). Application (resp. abstraction) is left (resp. right) associative. An abstraction with an empty set of binding symbols is written $\lambda_\varnothing\ p.u$. A λ-abstraction $\lambda x.t$ can be defined by $\lambda_{\{x\}}\ \widehat{x}.t$. The **identity function** $\lambda_{\{x\}}\ \widehat{x}.x$ is abbreviated I. The notation $|t|$ is used for the **size** of t, defined as expected.

A binding symbol $x \in \theta$ of an abstraction $\lambda_\theta\ p.s$ *binds* matchable occurrences of x in p and variable occurrences of x in s. The derived notions of **free variables** and **free matchables** are respectively denoted by $\mathtt{fv}(\_)$ and $\mathtt{fm}(\_)$. This is illustrated in Figure 3.6.

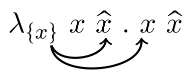


Figure 3.6: Binding in PPC

Formally, **free variables** and **free matchables** of terms are defined by: $\mathtt{fv}(x) := \{x\}$, $\mathtt{fv}(\widehat{x}) := \varnothing$, $\mathtt{fv}(tu) := \mathtt{fv}(t) \cup \mathtt{fv}(u)$, $\mathtt{fv}(\lambda_\theta\ p.u) := (\mathtt{fv}(u)\backslash\theta) \cup \mathtt{fv}(p)$, $\mathtt{fm}(x) := \varnothing$, $\mathtt{fm}(\widehat{x}) := \{x\}$, $\mathtt{fm}(tu) := \mathtt{fm}(t) \cup \mathtt{fm}(u)$, $\mathtt{fm}(\lambda_\theta\ p.u) := (\mathtt{fm}(p)\backslash\theta) \cup \mathtt{fm}(u)$.

As usual, we consider terms up to **alpha-conversion**, i.e. up to renaming of bound matchables and variables. A **constructor** is a matchable occurring in a term, such that all its occurrences are free. To ease the presentation, they are often denoted in typewriter fonts $\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \ldots$, thus for example $\lambda_{\{x,y\}}\ \widehat{x}\ y\ \mathtt{a}.y$ denotes $\lambda_{\{x,y\}}\ \widehat{x}\ y\ \widehat{z}.y$. The distinction between matchables and variables is unnecessary for standard (static) patterns which do not contain free variables.

A **position** is either ϵ (the empty position), or na, where $n \in \{1,2\}$ and a is a position. We use $a, b, \ldots$ (resp. $\mathcal{A}, \mathcal{B}, \ldots$ and $\delta, \rho, \pi, \ldots$) to denote positions (resp. sets and sequences of positions) and $b\mathcal{A}$ to mean $\{ba \mid a \in \mathcal{A}\}$. The **set** $\mathtt{Pos}(t)$ **of positions** of t is defined as expected, provided that for abstractions $\lambda_\theta\ p.s$ positions inside both p and s are considered. Formally, $\mathtt{Pos}(x) = \mathtt{Pos}(\widehat{x}) = \{\epsilon\}$, $\mathtt{Pos}(tu) = \{\epsilon\} \cup \{1a \ /\ a \in \mathtt{Pos}(t)\} \cup \{2a \ /\ a \in \mathtt{Pos}(u)\}$, and $\mathtt{Pos}(\lambda_\theta p.s) = \{\epsilon\} \cup \{1a \ /\ a \in \mathtt{Pos}(p)\} \cup \{2a \ /\ a \in \mathtt{Pos}(s)\}$. Here is an example $\mathtt{Pos}(\lambda_{\{x\}}\ \mathtt{a}\ \mathtt{b}.\mathtt{a}\ x\ x) = \{\epsilon, 1, 2, 11, 12, 21, 22, 211, 212\}$.

We write $t|_a$ for the **subterm of** t **at position** a and $t[s]_a$ for the **replacement** of the subterm at position a in t by s. Notice that replacement may capture variables. An *occurrence* of a term s in a term t is any position $p \in \mathtt{Pos}(t)$ verifying $t\,|_p = s$. Particularly, *variable occurrences* are defined this way.

We write $a \leqslant b$ (resp. $a \parallel b$) when the position a is a **prefix** of (resp. **disjoint** from) the position b. Notice that $a \parallel b$ and $a \leqslant c$ implies $c \parallel b$. All these notions are defined as expected [BN98] and extended to sets of positions as well. Particularly, given a position a and a set of positions $\mathcal{B}$, we will say that $a \leqslant \mathcal{B}$ iff $a \leqslant b$ for all $b \in \mathcal{B}$, and analogously for $<$, $\parallel$, etc.. Finally, we write $s \subseteq t$ if s is a subterm of t (note in particular $s \subseteq s$).

A **substitution** σ is a mapping from variables to terms with finite domain $\mathtt{dom}(\sigma)$. We write $\{x_1 \to t_1, \ldots, x_n \to t_n\}$ for a substitution with domain $\{x_1, \ldots, x_n\}$. The set of free variables of a substitution σ are defined as follows: $\mathtt{fv}(\sigma) = \bigcup_{x \in \mathtt{dom}(\sigma)} \mathtt{fv}(\sigma x)$. Similarly for $\mathtt{fm}(\sigma)$. The **symbols** of σ are $\mathtt{sym}(\sigma) := \mathtt{dom}(\sigma) \cup \mathtt{fv}(\sigma) \cup \mathtt{fm}(\sigma)$. A set of symbols θ **avoids** a substitution σ, written $\theta\#\sigma$, iff $\theta \cap \mathtt{sym}(\sigma) = \varnothing$. The **application of a substitution** σ to a term is written and defined as usual on alpha-equivalence classes; in particular $\sigma(\lambda_\theta\ p.s) := \lambda_\theta\ \sigma(p).\sigma(s)$, if $\theta\#\sigma$. Notice that data structures and matchable forms are stable by substitution. The **restriction** of a substitution σ to a set of variables $\{x_1, \ldots, x_n\} \subseteq \mathtt{dom}(\sigma)$ is written $\sigma|_{\{x_1,\ldots,x_n\}}$. The **composition** $\sigma \circ \eta$ of two substitutions σ and η is defined by $(\sigma \circ \eta)x = \sigma(\eta x)$.

The following notation is useful to define the reduction strategy $\mathcal{S}$, and later to prove properties about it, in Section 3.5.

Notation 3.4.1. *If t and θ are a term and a set of symbols respectively, then $bm(t, \theta)$ denotes the predicate which is true iff t* ***is a matchable bound*** *by θ, i.e., if $t = \widehat{x}$ for some $x \in \theta$.*

Matching and Semantics.
The definition of the rewrite rule of PPC resorts to the notions of *match* and *matching operation.*

Definition 3.4.2. *A* ***match*** *μ is either a substitution or a special constant in the set $\{\texttt{fail}, \texttt{wait}\}$. A* ***match*** *is* ***positive*** *if it is a substitution; it is* ***decided*** *if it is either positive or* `fail`*.*

The notions of domain, free variables and free matchables extends to matches as follows: $\texttt{dom}(\texttt{fail}) = \texttt{fv}(\texttt{fail}) = \texttt{fm}(\texttt{fail}) = \varnothing$*, while* $\texttt{dom}(\texttt{wait})$, $\texttt{fv}(\texttt{wait})$ *and* $\texttt{fm}(\texttt{wait})$ *are undefined. The restriction to a set of variables is also extended to matches, by defining* $\texttt{wait}|_{\{x_1,\ldots,x_n\}} = \texttt{wait}$ *and* $\texttt{fail}|_{\{x_1,\ldots,x_n\}} = \texttt{fail}$*, for any set of variables* $\{x_1, \ldots, x_n\}$*. Furthermore, the composition is extended to matches, as follows. If μ_1 and μ_2 are matches of which at least one is* `fail`*, then $\mu_2 \circ \mu_1$ is defined to be* `fail`*. Otherwise, if μ_1 and μ_2 are matches of which at least one is* `wait`*, then $\mu_2 \circ \mu_1$ is defined to be* `wait`*. Thus, in particular,* $\texttt{fail} \circ \texttt{wait}$ *is* `fail`*.*

Definition 3.4.3. *The* ***application of a match*** *μ to a term t, written μt, is defined as follows: if μ is a substitution, then it is applied as explained above; if $\mu = \texttt{wait}$, then μt is undefined; if $\mu = \texttt{fail}$, then μt is the identity function I.*

Other *closed terms in normal form* could be taken to define the result of $\texttt{fail}(t)$. The choice of I prevents computation after a matching failure to be blocked, and moreover allows to encode pattern-matching definitions given by alternatives [JK09], without the need of additional constructs.

The **disjoint union** of matches is a crucial operation in the definition of the meaning of matching in PPC.

Definition 3.4.4. *The* ***disjoint union*** *of two matches μ_1 and μ_2, notation $\mu_1 \uplus \mu_2$ is defined as: their union if both μ_i are substitutions and* $\texttt{dom}(\mu_1) \cap \texttt{dom}(\mu_2) = \varnothing$*;* `wait` *if either of the μ_i is* `wait` *and none is* `fail`*;* `fail` *otherwise.*

This definition of disjoint union of matches validates the following equations which are responsible for the non-sequential nature of PPC, as we will discuss in Section 3.4.2:

$$\texttt{fail} \uplus \texttt{wait} = \texttt{wait} \uplus \texttt{fail} = \texttt{fail}$$

Now we define how the matching operation is modeled in PPC.

Definition 3.4.5. *The* ***compound matching operation*** *takes a term, a set of binding symbols and a pattern and returns a match, it is defined by applying the following equations in order:*

$$\begin{array}{rcll}
\{\!\!\{\widehat{x} \rhd_\theta t\}\!\!\} & := & \{x := t\} & \text{if } x \in \theta \\
\{\!\!\{\widehat{x} \rhd_\theta \widehat{x}\}\!\!\} & := & \{\} & \text{if } x \notin \theta \\
\{\!\!\{pq \rhd_\theta tu\}\!\!\} & := & \{\!\!\{p \rhd_\theta t\}\!\!\} \uplus \{\!\!\{q \rhd_\theta u\}\!\!\} & \text{if } tu, pq \in \boldsymbol{MF} \\
\{\!\!\{p \rhd_\theta t\}\!\!\} & := & \texttt{fail} & \text{if } p, t \in \boldsymbol{MF} \\
\{\!\!\{p \rhd_\theta t\}\!\!\} & := & \texttt{wait} & \text{otherwise}
\end{array}$$

The name "compound" given to this operation is related to the third clause, where the matching of a compound argument w.r.t. a compound pattern is specified. The use of disjoint union in that case of the previous definition restricts positive compound matching to linear patterns;[5] disjoint union of two substitutions fails whenever their domains are not disjoint. Notice also the restriction to *matchable forms* in that clause and in the following one. These features are necessary to guarantee confluence.

Definition 3.4.6. *The result of the* ***matching operation***[6] $\{p/_\theta\ t\}$ *is defined to be the* check *of* $\{\!\{p \rhd_\theta t\}\!\}$ *on* θ*; where the* ***check*** *of a match* μ *on* θ *is* `fail` *if* μ *is a substitution whose domain is not* θ*,* μ *otherwise.*

Definition 3.4.7. *A redex* $(\lambda_\theta p.s)u$ *where* $\{p/_\theta\ u\} = \texttt{fail}$ *is called a* ***matching failure****.*

The previous definitions allow to introduce the only rewrite rule of PPC.

Definition 3.4.8. *The* ***reduction relation*** *of PPC is generated by the rule:*

$$(\lambda_\theta p.s)u \ \rightarrow\ \{p/_\theta\ u\}s \qquad \text{if } \{p/_\theta\ u\} \text{ is decided}$$

where $\{p/_\theta\ u\}$ *denotes the PPC* matching operation*, that is, the meaning in this calculus of the* matching *of the argument* u *w.r.t. the pattern* p*.*

In the just introduced rule, the matching of the argument u w.r.t. the pattern p is defined by means of the matching operation $\{p/_\theta\ u\}$. The result of this operation, a match, is applied to the argument s. In fact, a match is applied only if it is decided, i.e. if it is a substitution or the constant `fail`. If the match is `wait`, then the rule does not apply.

We give some examples of matching and reduction steps, according to the just given definitions. The match $\{\texttt{a}\,\widehat{x}\,\widehat{y}/_{\{x,y\}}\ \texttt{a}\,\texttt{b}\,(I\texttt{a})\}$ yields the substitution $\{x \rightarrow \texttt{b}, y \rightarrow I\texttt{a}\}$. In turn, $\{\texttt{a}\,\widehat{x}\,\widehat{y}/_{\{x,y\}}\ \texttt{c}\,\texttt{b}\,(I\texttt{a})\} = \texttt{fail}$, since `a` and `c` are different constructors, and $\{\texttt{a}\,\widehat{x}\,\widehat{y}/_{\{x,y\}}\ I\texttt{b}\} = \texttt{wait}$, because the term $I\texttt{b}$ is not a matchable form. Therefore, the following are valid reduction steps in PPC:

$$(\lambda_{\{x,y\}}\texttt{a}\,\widehat{x}\,\widehat{y}.yx)(\texttt{a}\,\texttt{b}\,(I\texttt{a})) \ \longrightarrow\ I\,\texttt{a}\,\texttt{b} \qquad\qquad (\lambda_{\{x,y\}}\texttt{a}\,\widehat{x}\,\widehat{y}.yx)(\texttt{c}\,\texttt{b}\,(I\texttt{a})) \ \longrightarrow\ I$$

while the rewrite rule does not apply to $(\lambda_{\{x,y\}}\texttt{a}\,\widehat{x}\,\widehat{y}.yx)(I\texttt{b})$. Notice that the following reduction sequence can be constructed from the latter term:

$$(\lambda_{\{x,y\}}\texttt{a}\,\widehat{x}\,\widehat{y}.yx)(I\texttt{b}) \ \longrightarrow\ (\lambda_{\{x,y\}}\texttt{a}\,\widehat{x}\,\widehat{y}.yx)\texttt{b} \ \longrightarrow\ I$$

Other matching examples follow: $\{\widehat{x}\widehat{x}/_{\{x\}}\ uv\}$ gives `fail` because $\widehat{x}\widehat{x}$ is not linear; $\{\widehat{x}\widehat{y}/_{\{x,y,z\}}\ uv\}$ gives `fail` because $\{x,y,z\} \neq \{x,y\}$, $\{\widehat{x}/_\varnothing\ u\}$ gives `fail` because $\varnothing \neq \{x\}$; $\{\widehat{y}/_{\{x\}}\ \widehat{y}\}$ gives `fail` because $\{x\} \neq \varnothing$; $\{\widehat{x}\widehat{y}/_{\{x\}}\ u\widehat{z}\}$ gives `fail` because $\{\!\{\widehat{y} \rhd_{\{x\}} \widehat{z}\}\!\}$ is `fail`; $\{\widehat{x}\widehat{y}/_\varnothing\ u\widehat{z}\}$ gives `fail` for the same reason.

[5]A pattern p is linear w.r.t. θ if for every x in θ, the matchable $\widehat{x}$ appears at most once in p.

[6]Note that our notation for (compound) matching differs from [JK06a] and [JK09]: the pattern and argument appear in reversed order there.

3.4.2 Non-sequentiality in PPC

Let us consider the term:

$$t = (\lambda_{\{x\}}\texttt{p}\,\widehat{x}\,\texttt{m}\,\texttt{s}.x)\ (\texttt{p}\,(I\texttt{a})\,(I\texttt{f})\,(I\texttt{d}))$$

corresponding to the non-sequentiality example in the introduction of this chapter. To verify that the rewrite rule of PPC does not apply to this term as a whole, observe the definition of compound matching yields:

$$\begin{array}{l}\{\!\{\texttt{p}\,\widehat{x}\,\texttt{m}\,\texttt{s} \rhd_{\{x\}} \texttt{p}\,(I\texttt{a})\,(I\texttt{f})\,(I\texttt{d}))\}\!\} \\ \quad = \{\!\{\texttt{p} \rhd_{\{x\}} \texttt{p}\}\!\} \uplus \{\!\{\widehat{x} \rhd_{\{x\}} I\texttt{a}\}\!\} \uplus \{\!\{\texttt{m} \rhd_{\{x\}} I\texttt{f}\}\!\} \uplus \{\!\{\texttt{s} \rhd_{\{x\}} I\texttt{d}\}\!\} \\ \quad = \varnothing \uplus \{x \to I\texttt{a}\} \uplus \texttt{wait} \uplus \texttt{wait} \\ \quad = \texttt{wait}\end{array}$$

Changing *either* of the occurrences of wait in $\varnothing \uplus \{x \to I\texttt{a}\} \uplus \texttt{wait} \uplus \texttt{wait}$ to fail would cause the result of the compound matching to fail. This is a consequence of how disjoint union is defined on matches, and particularly of the equations $\texttt{fail} \uplus \texttt{wait} = \texttt{wait} \uplus \texttt{fail} = \texttt{fail}$.

In turn, the contraction of $I\texttt{f}$ or $I\texttt{d}$ in t would imply the third, resp. fourth, component of $\varnothing \uplus \{x \to I\texttt{a}\} \uplus \texttt{wait} \uplus \texttt{wait}$ to change from wait to fail. We check this assertion for the former case, the latter being analogous.

$$\begin{array}{l}\{\!\{\texttt{p}\,\widehat{x}\,\texttt{m}\,\texttt{s} \rhd_{\{x\}} \texttt{p}\,(I\texttt{a})\,\texttt{f}\,(I\texttt{d}))\}\!\} \\ \quad = \{\!\{\texttt{p} \rhd_{\{x\}} \texttt{p}\}\!\} \uplus \{\!\{\widehat{x} \rhd_{\{x\}} I\texttt{a}\}\!\} \uplus \{\!\{\texttt{m} \rhd_{\{x\}} \texttt{f}\}\!\} \uplus \{\!\{\texttt{s} \rhd_{\{x\}} I\texttt{d}\}\!\} \\ \quad = \varnothing \uplus \{x \to I\texttt{a}\} \uplus \texttt{fail} \uplus \texttt{wait} \\ \quad = \texttt{fail}\end{array}$$

Hence the possibility of the two reduction sequences:

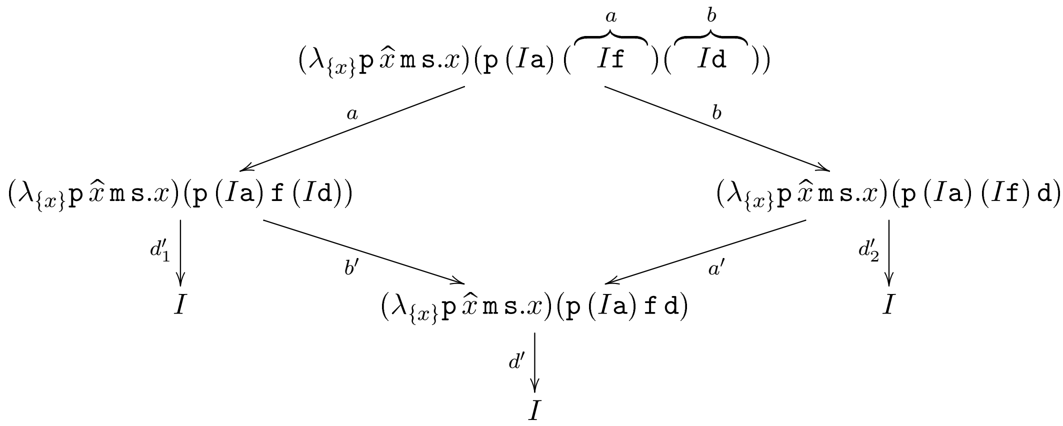
$$\begin{array}{lclcl}(\lambda_{\{x\}}\texttt{p}\,\widehat{x}\,\texttt{m}\,\texttt{s}.x)\ (\texttt{p}\,(I\texttt{a})\,(I\texttt{f})\,(I\texttt{d})) & \to & (\lambda_{\{x\}}\texttt{p}\,\widehat{x}\,\texttt{m}\,\texttt{s}.x)\ (\texttt{p}\,(I\texttt{a})\,\texttt{f}\,(I\texttt{d})) & \to & I \\ (\lambda_{\{x\}}\texttt{p}\,\widehat{x}\,\texttt{m}\,\texttt{s}.x)\ (\texttt{p}\,(I\texttt{a})\,(I\texttt{f})\,(I\texttt{d})) & \to & (\lambda_{\{x\}}\texttt{p}\,\widehat{x}\,\texttt{m}\,\texttt{s}.x)\ (\texttt{p}\,(I\texttt{a})\,(I\texttt{f})\,\texttt{d}) & \to & I\end{array}$$

which testify that none of the steps in t is needed, and hence that PPC is non-sequential.

The example shows that the ability of handling *dynamic patterns*, i.e. to perform reduction steps inside the pattern of an abstraction, is not crucial for the non-sequential nature of PPC. Non-sequentiality stems from the error mechanism of the calculus, which applies also to static, algebraic patterns.

Sequentiality of PPC can be recovered (see e.g. [Jay09, Bal10a, Bal10b]) by simplifying the equations of disjoint union, however some meaningful terms will no longer be normalising. E.g., if $\texttt{fail} \uplus \mu$ is defined to be fail, while $\texttt{wait} \uplus \texttt{fail} = \texttt{wait}$ and $\sigma \uplus \texttt{fail} = \texttt{fail}$, then $(\lambda_{\varnothing}\ \texttt{a}\ \texttt{b}\ \texttt{b}\ .\widehat{y})\ (\texttt{a}\ \Omega\ \texttt{c})$, where Ω is a non-terminating term, would never fail as expected.

Finally, we want to remark that the example developed in this section shows that PPC does not enjoy the Stability axiom. The counterexample shown in Section 2.1.4, based on the "parallel-or" rewriting system, can be rephrased as follows



The contraction of either of the disjoint steps a and b suffices to create the external matching failure; hence the counterexample to the $\mathsf{Stability}$ axiom.

3.4.3 PPC as an ARS

Let us define an ARS to describe PPC; we give some examples later on.

Definition 3.4.9. *The ARS* $\mathfrak{A}_{PPC} = \langle \mathcal{O}, \mathcal{R}, \mathsf{src}, \mathsf{tgt}, [\![\cdot]\!], <, \ll \rangle$ *is defined as follows.*

Objects

The set $\mathcal{O}$ *of* objects *is the set of terms of PPC.*

Steps, source, target

A step *is any pair* $\langle t, a\rangle$*, such that* $t|_a = (\lambda_\theta p.s)u$ *and* $\{p/_\theta\ u\}$ *is decided. In this case* $\mathsf{src}(\langle t, a\rangle) := t$ *and* $\mathsf{tgt}(\langle t, a\rangle) := t[\{p/_\theta\ u\}s]_a$*.*

Residual relation

If $a_r = \langle t, a\rangle$*,* $b_r = \langle t', b\rangle$ *and* $b'_r = \langle u, b'\rangle$ *are steps, then* $b_r[\![a_r]\!]b'_r$ *iff* $t' = t$*,* $u = \mathsf{tgt}(a_r)$*, and one of the following cases apply, where* $t|_a = (\lambda_\theta p.s)u$*:*

- $a \nleq b$ *and* $b' = b$*.*
- $b = a12n$*,* $b' = an$ *and* $\{p/_\theta\ u\} \neq \mathtt{fail}$*.*
- $b = a2mn$*,* $b' = akn$*,* $\{p/_\theta\ u\} \neq \mathtt{fail}$*, and there is a variable* $x \in \theta$ *such that* $t|_{a11m} = p|_m = \widehat{x}$ *and* $t|_{a12k} = s|_k = x$*.*

Embedding

Let $a_r = \langle t, a\rangle$ *and* $b_r = \langle t, b\rangle$ *be steps. We define* $a_r < b_r$ *iff* $a < b$*. Given* $c_r = \langle t, c\rangle$*, notice that whenever* $a_r < c_r$ *and* $b_r < c_r$*, then* a_r *and* b_r *are comparable w.r.t. the embedding, i.e. either* $a_r = b_r$*,* $a_r < b_r$ *or* $b_r < a_r$*.*

Gripping

Let $a_r = \langle t, a\rangle$ *and* $b_r = \langle t, b\rangle$ *be steps and let* $t|_a = (\lambda_\theta p.s)u$*. Then* $a_r \ll b_r$ *iff* $\{p/_\theta\ u\} \neq \mathtt{fail}$*,* $b = a12n$*, and* $\theta \cap \mathtt{fv}(s|_n) \neq \emptyset$*.*

Notation 3.4.10. *Given a step* $\langle t, a\rangle$ *we will often denote it* a_r*; this notation shall prove convenient when we address the compliance of PPC w.r.t. the axioms of an ARS. This notational convention is extended to multisteps: if* A *is a set of redex positions in the term* t*, we will use* A_r *to denote* $\{\langle t, a\rangle \ /\ a \in A\}$*. This extension is used in Sections 3.5.1 and 3.5.2.*

Using positions to identify redexes yields the given, somewhat complicated, definitions of the residual and gripping relations. The former aims to trace redexes from source term to target term, while the latter characterises, by means of positions, the idea of gripping described in Section 1.3.1. The use of one common notion, namely that of positions, to define all the relations, eases the proofs of results where several of these relations are involved. Hence the reason why we favor positions over other mechanisms, such as *labeling* which is used for $\lambda_{\widetilde{\mathtt{lsub}}}$ in Chapter 4 and also for λ-calculus in e.g. [Bar84, Kri90].

To exemplify the definitions of residuals and gripping, consider the following step:

$$t = (\lambda_{\{x,y\}} \mathtt{c}\widehat{x}\widehat{y}.x\,(Ix))\,(\mathtt{c}(\mathtt{d}(I\mathtt{a}))(I\mathtt{b}))\,(I\mathtt{e}) \quad\to\quad \mathtt{d}(I\mathtt{a})\,(I(\mathtt{d}(I\mathtt{a})))\,(I\mathtt{e}) = u$$

where the contracted redex is $a_r = \langle t, 1\rangle$. Let us analyse the residuals of each step in t:

- Take $\langle t, 1122\rangle$, whose subterm is Ix. The second clause of the residual definition applies, where $n = 2$. Therefore there is one residual, namely $\langle u, 12\rangle$.

- Take $\langle t, 12122\rangle$, whose subterm is $I\mathtt{a}$. The third clause applies, where $m = 12$ and $n = 2$; observe $\mathtt{c}\widehat{x}\widehat{y}|_{12} = \widehat{x}$. There are two occurrences of x in $x(Ix)$, at positions 1 and 22, these are the possible k in the definition of residuals. Hence there are two residuals, $\langle u, 112\rangle$ and $\langle u, 1222\rangle$.

- Take $\langle t, 122\rangle$, whose subterm is $I\mathtt{b}$. The third clause applies, where $m = 2$ and $n = \epsilon$; observe $\mathtt{c}\widehat{x}\widehat{y}|_2 = \widehat{y}$. There are no occurrences of y in $x(Ix)$, hence there are no residuals of this step after a_r.

- Take $\langle t, 2\rangle$, whose subterm is $I\mathtt{e}$. The first clause applies, then there is one residual: $\langle u, 2\rangle$.

The only pair in the gripping relation among the residuals of t is $a_r \ll \langle t, 1122\rangle$. W.r.t. the definition of gripping, we have $n = 2$ and $x \in \{x, y\} \cap \mathtt{fv}(Ix)$.

Let us consider, as a second example, this step:

$$t = (\lambda_{\{x,y\}} \mathtt{c}\widehat{x}\widehat{y}.x\,(Ix))\,(\mathtt{d}(I\mathtt{a}))\,(I\mathtt{e}) \quad\to\quad I\,(I\mathtt{e}) = u$$

and define $a_r = \langle t, 1\rangle$. Observe that there is no copy, in u, of any of the redexes embedded by a_r in t. This observation motivates the condition $\{p/_\theta\ u\} \neq \mathtt{fail}$ in the second and third clauses in the definition of residuals.

For a more involved example about gripping, we consider this term:

$$t = \underbrace{\big(\lambda_{\{z\}}\widehat{z}.\,\underbrace{(\lambda_{\{x,y\}}\mathtt{c}\widehat{x}\widehat{y}.(\underbrace{I(\mathtt{a}x)}_{c_r})(\underbrace{I(zy)}_{d_r})(\underbrace{Iz}_{e_r}))\,(\mathtt{ccc})}_{b_r}\;z\big)\mathtt{d}}_{a_r}$$

where the redexes are underlined. We have $b_r \ll c_r$, since $x \in \{x, y\} \cap \mathtt{fv}(I(\mathtt{a}x))$. Analogously, we have $b_r \ll d_r$. On the other hand, $b_r \not\ll e_r$, because $\{x, y\} \cap \mathtt{fv}(Iz) = \varnothing$. Moreover, observe that c_r, d_r and e_r are inside the body of a_r as well. Observing the occurrences of z, we obtain $a_r \not\ll c_r$, $a_r \ll d_r$ and $a_r \ll e_r$. Finally, we remark that there is another pair in the gripping relation for this term: $a_r \ll b_r$.

In the remainder of this section, we verify that the ARS modeling PPC verifies the fundamental axioms, FD, SO, the embedding axioms except for Stability, the gripping axioms, and also Pivot.

Initial axioms.

Self Reduction is immediate from the definition of residuals for PPC: none of the cases there applies for $a_r[\![a_r]\!]$. Finite Residuals follows from the fact that terms are finite. Axiom Ancestor Uniqueness is proved below.

Lemma 3.4.11 (Ancestor Uniqueness). *Let $b_{r_1}, b_{r_2}, a_r, b'_r$ be steps verifying $b_{r_1}[\![a_r]\!]b'_r$ and $b_{r_2}[\![a_r]\!]b'_r$. Then $b_{r_1} = b_{r_2}$.*

Proof. Let $b_{r_1} = \langle t', b_1\rangle$ and $b_{r_2} = \langle t', b_2\rangle$ where $t' = \mathtt{tgt}(a_r)$. We prove that $b_1 = b_2$. Let $t|_a = (\lambda_\theta p.s)u$. We consider three cases according to the definition of $b_{r_1}[\![a_r]\!]b'_r$.

- If $a \nleq b_1$, then $b_1 = b'$ so that $a \nleq b'$. A straightforward case analysis on the definition of residuals yields $a \nleq b_2$, therefore $b_1 = b_2 = b'$.
- If $b_1 = a2mn$ and $b' = akn$, then $s|_k = x$ and $p|_m = \widehat{x}$ for some $x \in \theta$. Observe that $a < b'$ implies $a < b_2$. We consider two cases. If $b_2 = a12n'$ and $b' = an'$, then $kn = n'$. This would imply $t|_{b_2} = s|_{kn}$ has the form $(\lambda_{\theta'}p'.s')u'$, contradicting $s|_k$ being a variable. Therefore, $akn = b' = ak'n'$ and $b_2 = a2m'n'$, where $s|_{k'} = y$ and $p|_{m'} = \widehat{y}$ for some $y \in \theta$. Observe that $k < k'$, i.e. $k' = kc$ where $c \neq \epsilon$, would imply $kc \in \mathtt{Pos}(s)$, contradicting the fact that $s|_k$ is a variable; so that $k \nless k'$. We obtain $k' \nless k$ analogously. On the other hand, $k \parallel k'$ would contradict $kn = k'n'$. Hence $k = k'$, implying $n = n'$ and also $y = x$. In turn, $\{p/_\theta\ u\}$ being positive implies that p is linear, and then $m = m'$. Thus we conclude.
- If $b_1 = a12n$ and $b' = an$, then we have again that $a < b'$ implies $a < b_2$. On the other hand, assuming $b_2 = a2m'n'$, so that $an = b' = akn'$, would yield a contradiction as already stated. Therefore $b_2 = a12n'$ and $an = b' = an'$, implying $n = n'$ and consequently $b_1 = b_2$. □

Finally, FD and SO are left for the end of this section.

The Enclave–Creation axiom.

To verify Enclave–Creation involves a rather long technical development, including some preliminary lemmas, particularly a creation lemma indicating the creation cases for PPC. One of these lemmas is used in the proof of subsequent axioms as well.

Lemma 3.4.12. *Let $p \twoheadrightarrow p'$ and $u \twoheadrightarrow u'$. Then,*

(i) $\{\!\{p \rhd_\theta u\}\!\}$ *positive implies* $\{\!\{p' \rhd_\theta u'\}\!\}$ *positive,*

(ii) $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{fail}$ *implies* $\{\!\{p' \rhd_\theta u'\}\!\} = \mathtt{fail}$.

(iii) $\{p/_\theta\ u\}$ *positive implies* $\{p'/_\theta\ u'\}$ *positive,*

(iv) $\{p/_\theta\ u\} = \mathtt{fail}$ *implies* $\{p'/_\theta\ u'\} = \mathtt{fail}$.

Proof. We prove item (i). Given $\{\!\{p \rhd_\theta u\}\!\}$ is positive, a straightforward induction on p yields that p is a normal form, implying $p' = p$. If $bm(p, \theta)$, then $\{\!\{p \rhd_\theta u'\}\!\}$ is positive for any term u'. If p is a matchable and $\neg bm(p, \theta)$, then $\{\!\{p \rhd_\theta u\}\!\}$ positive implies $u = p$, i.e. u is a normal form, and therefore $u' = u$, which suffices to conclude. Assume $p = p_1p_2$. Then the hypotheses imply $p \in \boldsymbol{MF}$, $u = u_1u_2 \in \boldsymbol{MF}$, and $\{\!\{p_i \rhd_\theta u_i\}\!\}$ positive for $i = 1, 2$. In turn, $u \in \boldsymbol{MF}$ implies $u' = u'_1u'_2$ and $u_i \twoheadrightarrow u'_i$ for $i = 1, 2$. Hence, the

IH can be applied for each $u_i \twoheadrightarrow u'_i$, which suffices to conclude. Finally, any other case would contradict $\{\!\{p \rhd_\theta u\}\!\}$ positive.

We prove item (ii). Observe $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{fail}$ implies $p, u \in \boldsymbol{MF}$, and therefore $p', u' \in \boldsymbol{MF}$. Therefore, p and p' share their syntactic form (i.e. they are either both matchables, both applications or both abstractions), and similarly for u and u'. If p and u, and therefore p' and u', have different syntactic forms, or else if p, p', u, u' are abstractions, then it suffices to observe that $\{\!\{p' \rhd_\theta u'\}\!\} = \mathtt{fail}$ for any such p' and u'. If p, p', u, u' are matchables, then $p = p'$ and $u = u'$, thus we immediately conclude. Assume $p = p_1p_2$, $p' = p'_1p'_2$, $u = u_1u_2$ and $u' = u'_1u'_2$. In this case, hypotheses imply $\{\!\{p_i \rhd_\theta u_i\}\!\} = \mathtt{fail}$ for some $i \in \{1, 2\}$, and moreover $p, u \in \boldsymbol{MF}$ imply $p_i \twoheadrightarrow p'_i$ and $u_i \twoheadrightarrow u'_i$. Therefore, we conclude by applying the IH, and recalling that $\mathtt{fail} \uplus R = \mathtt{fail}$ for any possible R.

To prove items (iii) and (iv), we observe that a straightforward induction on p yields that $\{\!\{p \rhd_\theta u\}\!\} = \sigma$ implies $\mathtt{dom}(\sigma) = \mathtt{fm}(p)$, and therefore in this case $\{p/_\theta\ u\}$ is positive iff $\theta = \mathtt{fm}(p)$, and $\{p/_\theta\ u\} = \mathtt{fail}$ otherwise. Recall also that $\{\!\{p \rhd_\theta u\}\!\}$ positive implies p being a normal form, and then $p' = p$. For item (iii): $\{p/_\theta\ u\}$ positive implies $\{\!\{p \rhd_\theta u\}\!\} = \sigma$ where $\theta = \mathtt{fm}(p) = \mathtt{fm}(p')$. On the other hand, item (i) just proved implies $\{\!\{p' \rhd_\theta u'\}\!\} = \sigma'$, which suffices to conclude. For item (iv): assume $\{p/_\theta\ u\} = \mathtt{fail}$. If $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{fail}$, then item (ii) just proved implies $\{\!\{p' \rhd_\theta u'\}\!\} = \mathtt{fail}$, thus we conclude. Otherwise, $\{\!\{p \rhd_\theta u\}\!\} = \sigma$ and $\sigma \neq \mathtt{fm}(p) = \mathtt{fm}(p')$, and item (i) just proved implies $\{\!\{p' \rhd_\theta u'\}\!\} = \sigma'$, which suffices to conclude. □

Lemma 3.4.13 (Creation cases)**.** *Let $t \xrightarrow{a_r} t'$, and $\varnothing[\![a_r]\!]b_r$, i.e. b_r is* created *by (the contraction of) a_r. Say $t|_a = (\lambda_\theta p.s)u$ and $t'|_b = (\lambda_{\theta'} p'.s')u'$. Then one of the following holds:*

Case I. *the contraction of a_r contributes to the creation of b_r from below, i.e., $b \in \mathtt{Pos}(t)$, $a = b1$ implying $t|_b = (\lambda_\theta p.s)uu'$, and either*

- *(i) $s = x$ where $x \in \theta$ and $\widehat{x}$ occurs in p, $\{p/_\theta\ u\} = \sigma$, $\sigma x = (\lambda_{\theta'} p'.s')$.*
- *(ii) $s = \lambda_{\theta'} p''.s''$, $\{p/_\theta\ u\} = \sigma$, $p' = \sigma p''$, $s' = \sigma s''$.*
- *(iii) $\{p/_\theta\ u\} = \mathtt{fail}$, $\lambda_{\theta'} p'.s' = I$.*

Case II. *the contraction of a_r contributes to the creation of b_r from above, i.e., $b = an$, $s|_n = xu''$, $\{p/_\theta\ u\} = \sigma$, $\sigma x = (\lambda_{\theta'} p'.s')$, $u' = \sigma u''$.*

Case III. *The argument of a redex pattern becomes decided. We have three such situations:*

- *(i) $b = an$, $s|_n = (\lambda_{\theta'} p''.s'')u''$, $\{p''/_\theta\ u''\} = \mathtt{wait}$, $\{p/_\theta\ u\} = \sigma$, $p' = \sigma p''$, $u' = \sigma u''$.*
- *(ii) $a = b2n$, $t|_b = (\lambda_{\theta'} p'.s')u''$, $\{p''/_\theta\ u''\} = \mathtt{wait}$.*
- *(iii) $a = b11n$, $t|_b = (\lambda_{\theta'} p''.s')u'$, $\{p''/_\theta\ u''\} = \mathtt{wait}$.*

Proof. We proceed by comparing a with b.

- If $a \parallel b$, then $t|_b = t'|_b$ so that $\langle t, b\rangle[\![a_r]\!]b_r$, contradicting the hypotheses.
- Assume $a \leqslant b$, i.e. $b = ac$.

  In this case, $\{p/_\theta\ u\} = \mathtt{fail}$ would imply $t'|_a = I$, contradicting $t'|_b$ being a redex. Then $\{p/_\theta\ u\} = \sigma$, implying $t'|_b = \sigma s|_c$. Observe that $c = kn$, $s|_k = x$

and $t'|_b = \sigma x|_n$ for some variable x would imply $\langle t, a2mn\rangle[\![a_r]\!]b_r$ where $p|_m = \widehat{x}$. Therefore $= s|_c = t_1 u''$ and $t'|_b = (\lambda_{\theta'} p'.s')u' = (\sigma t_1)\sigma u''$. If t_1 is a variable, so that $\sigma t_1 = \lambda_{\theta'} p'.s'$, then case II applies, otherwise case III.(i) applies.

- Assume $b < a$.

  If $a = b1$, i.e. $t|_b = (\lambda_\theta p.s)uu'$, then observe $\{p/_\theta\ u\}s = t'|_a = \lambda_{\theta'} p'.s'$. If $\{p/_\theta\ u\} =$ `fail`, then case I.(iii) applies. If s is a variable, then case I.(i) applies. Otherwise, s is an abstraction, so that case I.(ii) applies.

  If $b11 \leqslant a$, i.e. $t|_b = (\lambda_{\theta'} p''.s')u'$, then observe $\varnothing[\![a_r]\!]b_r$ implies $\{p''/_{\theta'}\ u'\} =$ `wait`. Then case III.(iii) applies. If $b2 \leqslant a$, a similar argument yields that case III.(ii) applies.

  Finally, $b12 \leqslant a$ implies $t|_b = (\lambda_{\theta'} p'.s'')u'$, and $t'|_b$ being a redex implies $\{p'/_{\theta'}\ u'\}$ decided so that $\langle t, b\rangle[\![a_r]\!]b_r$, contradicting the hypothesis. □

Lemma 3.4.14. *Let $t \xrightarrow{a_r} t'$ such that $t \notin \boldsymbol{MF}$ and $t' \in \boldsymbol{MF}$. Then a_r is outermost.*

Proof. By induction on t'.

If t' is a variable or a matchable, then $a = \epsilon$, thus we conclude.

If t' is an abstraction, then $a \neq \epsilon$ implies t is an abstraction contradicting $t \notin \boldsymbol{MF}$. Thus $a = \epsilon$ and we conclude.

If $t' = t'_1 t'_2$, then $t' \in \boldsymbol{MF}$ implies $t'_1 \in \boldsymbol{DS}$. We consider three cases. (i) If $a = \epsilon$ then we immediately conclude. (ii) If $2 \leqslant a$, then we contradict $t \notin \boldsymbol{MF}$. (iii) If $1 \leqslant a$, i.e. $a = 1a'$, then $t = t_1 t'_2$ and $t_1 \xrightarrow{a'_r} t'_1$. Observe that $t_1 \in \boldsymbol{DS}$ would contradict $t \notin \boldsymbol{MF}$, and $t_1 \in \boldsymbol{ABS}$ would imply $t'_1 \in \boldsymbol{ABS}$, contradicting $t'_1 \in \boldsymbol{DS}$. Therefore, $t_1 \notin \boldsymbol{MF}$, and hence the IH yields that $\langle t_1, a'\rangle$ is outermost. We conclude by observing that $t_1 \notin \boldsymbol{MF}$ implies that $\langle t, \epsilon\rangle$ is not a step. □

Lemma 3.4.15. *Let $t \xrightarrow{a_r} t'$ such that $\{\!\{p \rhd_\theta t\}\!\} =$ `wait` and $\{\!\{p \rhd_\theta t'\}\!\}$ is decided for some θ, p. Then a_r is outermost.*

Proof. We proceed by induction on t. Observe that $\{\!\{p \rhd_\theta t\}\!\} =$ `wait` implies $\neg bm(p, \theta)$. In turn, $\{\!\{p \rhd_\theta t'\}\!\}$ decided implies $p \in \boldsymbol{MF}$, and moreover $\neg bm(p, \theta)$ implies $t' \in \boldsymbol{MF}$. If $t \notin \boldsymbol{MF}$ then Lem. 3.4.14 suffices to conclude. Therefore, assume $t \in \boldsymbol{MF}$. In this case, $\{\!\{p \rhd_\theta t\}\!\} =$ `wait` implies $p = p_1 p_2$, $t = t_1 t_2$, and $\{\!\{p_i \rhd_\theta t_i\}\!\} \neq$ `fail` for $i = 1, 2$. Furthermore, $t \in \boldsymbol{MF}$ implies $a \neq \epsilon$. Assume $a = 1a'$, implying $t' = t'_1 t_2$ and $t_1 \xrightarrow{a'_r} t'_1$. Notice $\{\!\{p_1 \rhd_\theta t_1\}\!\}$ positive would imply $\{\!\{p_1 \rhd_\theta t'_1\}\!\}$ positive, cfr. Lem. 3.4.12, thus contradicting either $\{\!\{p \rhd_\theta t\}\!\} =$ `wait` (if $\{\!\{p_2 \rhd_\theta t_2\}\!\}$ is positive) or $\{\!\{p \rhd_\theta t'\}\!\}$ decided (if $\{\!\{p_2 \rhd_\theta t_2\}\!\} =$ `wait`); while $\{\!\{p_1 \rhd_\theta t'_1\}\!\} =$ `wait` would contradict $\{\!\{p \rhd_\theta t'\}\!\}$ decided. Hence IH can be applied to obtain $\langle t_1, a'\rangle$ outermost, which suffices to conclude (given $\langle t, \epsilon\rangle$ not being a step). The case $a = 2a'$ admits an analogous argument. □

Lemma 3.4.16. *Let $p \xrightarrow{a_r} p'$ such that $\{\!\{p \rhd_\theta t\}\!\} =$ `wait` and $\{\!\{p' \rhd_\theta t\}\!\}$ is decided for some θ, t. Then a_r is outermost.*

Proof. We proceed by induction on p. Observe that $\{\!\{p' \rhd_\theta t\}\!\}$ decided implies $p' \in \boldsymbol{MF}$. If $p \notin \boldsymbol{MF}$ then Lem. 3.4.14 suffices to conclude. Therefore, assume $p \in \boldsymbol{MF}$. This implies p' is not a matchable, and consequently $\{\!\{p' \rhd_\theta t\}\!\}$ decided implies $t \in \boldsymbol{MF}$. In turn, $\{\!\{p \rhd_\theta t\}\!\} =$ `wait` yields $t = t_1 t_2$, $p = p_1 p_2$, $\{\!\{p_i \rhd_\theta t_i\}\!\} \neq$ `fail` for $i = 1, 2$, and $a \neq \epsilon$.

Assume $a = 1a'$, implying $p' = p'_1 p_2$ and $p_1 \xrightarrow{a'_r} p'_1$. In this case, $\{\!\{p_1 \rhd_\theta t_1\}\!\}$ positive would imply p_1 to be a normal form, while $\{\!\{p'_1 \rhd_\theta t_1\}\!\} = \mathtt{wait}$ would contradict $\{\!\{p' \rhd_\theta t\}\!\}$ decided. Therefore the IH can be applied to obtain $\langle p_1, a' \rangle$ outermost, which suffices to conclude since $\langle p, \epsilon \rangle$ is not a step. The case $a = 2a'$ admits an analogous argument. □

Lemma 3.4.17 (Enclave–Creation). *Let a_r, b_r be steps such that $b_r < a_r$, $b_r[\![a_r]\!]b'_r$, and $\varnothing[\![a_r]\!]c'_r$. Then $b'_r < c'_r$.*

Proof. Observe that $a \nleqslant b$ implying $b' = b$. Say $t \xrightarrow{a_r} t'$, $t|_a = (\lambda_\theta p.s)u$, and $t'|_{c'} = (\lambda_{\theta'} p'.s')u'$. We proceed by case analysis w.r.t. Lem. 3.4.13.

Case I In this case $c' \in \mathtt{Pos}(t)$ and $a = c'1$, so that $t|_{c'} = (\lambda_\theta p.s)uu'$. Therefore, it suffices to observe that $b = c'$ would forbid b_r to be a step, then $b < a$ implies $b < c'$.

Cases II or III.(i) In either case $c' = an$, thus $b < a$ implies $b < c'$.

Case III.(ii) In this case $a = c'2n$ and $t|_{c'} = (\lambda_{\theta'} p'.s')u''$. Then, $b < a$ implies either $b < c'$, $b = c'$ or $b = c'2n'$ where $n' < n$. We conclude by observing that the second and third cases would contradict $\varnothing[\![a_r]\!]c'_r$ and Lem. 3.4.15 respectively.

Case III.(iii) In this case $a = c'11n$ and $t|_{c'} = (\lambda_{\theta'} p''.s')u'$. A similar analysis applies, resorting to Lem. 3.4.16 instead of Lem. 3.4.15. □

The other embedding and gripping axioms.

Linearity is immediate from the definition of residuals. The remaining embedding axioms, and also Grip–Instantiation, are related with the invariance of embedding w.r.t. residuals. The following lemma describes the exceptions to such invariance, so that several axioms can be obtained as simple corollaries of its statement.

Lemma 3.4.18. *Suppose $b_r[\![a_r]\!]b'_r$ and $c_r[\![a_r]\!]c'_r$, such that $\neg(b_r < c_r \leftrightarrow b'_r < c'_r)$. Then $(a_r < b_r) \wedge (a_r < c_r)$, and moreover*

- *either $(b_r \parallel c_r) \wedge (b'_r < c'_r) \wedge (a_r \ll b_r) \wedge (a2 \leqslant c)$,*
- *or $(b_r < c_r) \wedge (b'_r \parallel c'_r)$.*

Proof. Suppose $t \xrightarrow{a} u$ and $t|_a = (\lambda_\theta p.s)u$. If $\neg((a_r < b_r) \wedge (a_r < c_r))$, then we obtain $b_r < c_r \leftrightarrow b'_r < c'_r$ by the following case analysis:

- $a = b$ or $a = c$: either case would contradict the existence of b'_r and c'_r.
- $a \nleqslant b$ and $a \nleqslant c$: in this case $b' = b$ and $c' = c$, which suffices to conclude.
- $a \parallel b$ and $a < c$: implies $b \parallel c$ and $b' = b \parallel a \leqslant c'$, hence $b_r \nless c_r \;\wedge\; b'_r \nless c'_r$.
- $a < b$ and $a \parallel c$: analogous to the previous case.
- $b < a < c$: implies $b < c$ and $b' = b < a \leqslant c'$, hence $b_r < c_r \;\wedge\; b'_r < c'_r$.
- $c < a < b$: we obtain analogously $c < b$ and $c' < b'$, hence $b_r \nless c_r \;\wedge\; b'_r \nless c'_r$.

If $a < b$ and $a < c$, then we analyse the possible cases w.r.t. the residual relation, recalling that all cases suppose $\{p/_\theta\ u\} \neq \mathtt{fail}$, and therefore that p is linear.

- $b = a12n$ and $c = a12n'$. In this case $b' = an$ and $c' = an'$, thus we conclude immediately.

- $b = a12n$ and $c = a2m'n'$. In this case $b' = an$ and $c' = ak'n'$, where $p|_{m'} = \widehat{x}$ and $s|_{k'} = x$ for some $x \in \theta$. Observe $b \parallel c$. If $b' \nless c'$ then we conclude immediately, so that assume $b' < c'$, implying $n < k'n'$. Notice that $s|_n$ is a redex while $s|_{k'}$ is a variable, then $n < k'n'$ implies $n < k'$. Therefore $x \in \theta \cap \mathtt{fv}(s|_n)$, implying $a_r \ll b_r$. Thus we conclude.

- $b = a2mn$ and $c = a12n'$. In this case, $b' = akn$ and $c' = an'$, where $p|_m = \widehat{x}$ and $s|_k = x$ for some $x \in \theta$. Observe $b \parallel c$. Moreover $s|_{n'}$ being a redex while $s|_k$ is a variable implies $k \nless n'$, then $kn \nless n'$, hence $b' \nless c'$. Thus we conclude.

- $b = a2mn$ and $c = a2m'n'$. In this case $b' = akn$ and $c' = ak'n'$, where $p|_m = \widehat{x}$, $s|_k = x$, $p|_{m'} = \widehat{y}$ and $s|_{k'} = y$ for some $x, y \in \theta$. Both $s|_k$ and $s|_{k'}$ being variable occurrences implies $k = k'$ or $k \parallel k'$. An analogous argument yields $m = m'$ or $m \parallel m'$.

  If $b \nless c$ and $c \nless b$, then we conclude immediately.

  If $b < c$, then $m = m'$ implying $x = y$, and $n < n'$. If $k = k'$, then $b' < c'$, otherwise, $b' \parallel c'$. Thus we conclude.

  Finally, if $b' < c'$, then $k = k'$ and $n = n'$. But $k = k'$ implies $x = y$, and then $m = m'$ by linearity of p. Then $b < c$. □

It is easy to obtain Context-Freeness, Enclave–Embedding and Grip–Instantiation as corollaries of Lem. 3.4.18.

We verify Pivot; a previous lemma is needed first.

Lemma 3.4.19. *Let p, t, b such that $\{p/_\theta\ t\}$ is positive and $t|_b$ is a redex. Then there exists some $a \leqslant b$ verifying $p|_a = \widehat{x}$ for some $x \in \theta$.*

Proof. By induction on p, considering the cases in the definition of the compound matching allowing $\{p/_\theta\ t\}$ to be positive. If $bm(p, \theta)$ then taking $a = \epsilon$ allows to conclude. Otherwise, $\{p/_\theta\ t\}$ positive implies $p = p_1p_2$, $t = t_1t_2 \in \boldsymbol{MF}$, and $\{p_i/_\theta\ t_i\}$ positive for $i = 1, 2$. In turn, $t \in \boldsymbol{MF}$ and $t|_b$ being a redex imply $b \neq \epsilon$, then $b = ib'$ where $i \in \{1, 2\}$, hence $t|_b = t_i|_{b'}$. IH yields $p_i|_{a'} = \widehat{x}$ where $x \in \theta$, for some $a' \leqslant b'$. We conclude by taking $a = ia'$. □

Lemma 3.4.20 (Pivot)**.** *Let a_r, b_r, c_r, c'_r steps verifying $a_r < c_r$, $b_r < c_r$, $b_r \nleqslant a_r$, and $c_r[\![a_r]\!]c'_r$. Then there exists a step b'_r such that $b_r[\![a_r]\!]b'_r$ and $b'_r < c'_r$.*

Proof. Observe that $a < c$, $b < c$ and $b \nleqslant a$ implies $a < b < c$. We proceed by case analysis on the definition of residuals, considering $a_r < c_r$. Say $t|_a = (\lambda_\theta p.s)u$. Observe that $a < c$ and $c_r[\![a_r]\!]c'_r$ imply that $\{p/_\theta\ u\}$ is positive.

If $c = a12n'$, so that $c' = an'$, then $b < c$ implies $b = a12n$ and $n < n'$ (recall $t|_{a1} \in \boldsymbol{ABS}$). Hence, taking $b' = an$ suffices to conclude.

If $c = a2mn$, then $b = a2b''$ and $b'' < mn$. Observe that $p|_m = \widehat{x}$ where $x \in \theta$, and $c' = akn$ where $s|_k = x$. Lem. 3.4.19 implies $b'' = b_1b_2$ where $p|_{b_1} = \widehat{y}$. Notice that $b_1b_2 < mn$, along with both $p|_{b_1}$ and $p|_m$ being matchable occurrences, imply that $b_1 = m$, then $x = y$, and also $b_2 < n$. Hence we conclude by taking $b' = akb_2$. □

Finally, we verify the two remaining gripping axioms.

Lemma 3.4.21. *Let* $t = (\lambda_\theta p.s)u \xrightarrow{a_r} t'$, $c_r[\![a_r]\!]c'_r$, *and* $x \in \mathtt{fv}(t'|_{c'})$. *Then* $x \in \mathtt{fv}(t|_c)$, *or* $a_r \ll c_r$ *and* $x \in \mathtt{fv}(u)$.

Proof. By case analysis on the definition of residuals. If $a \not\leqslant c$ or $c = a2mn$, then $t|_c = t'|_{c'}$, implying $x \in t|_c$. Otherwise, i.e. if $c = a12n$, $c' = an$, and $\{p/_\theta\ u\} \neq \mathtt{fail}$, let us consider d such that $t'|_{c'd} = (\{p/_\theta\ u\}s)|_{nd} = x$. Given $n \in \mathtt{Pos}(s)$, it is easy to obtain $(\{p/_\theta\ u\}s)|_{nd} = (\{p/_\theta\ u\}(s|_n))|_d = (\{p/_\theta\ u\}(t|_c))|_d$. In turn, $x \in \mathtt{fv}(\{p/_\theta\ u\}(t|_c))$ yields easily $x \in \mathtt{fv}(t|_c)$ or $x \in \mathtt{fv}(u) \wedge t|_c \cap \theta \neq \varnothing$. We conclude by observing that the latter case implies $a_r \ll c_r$. □

Lemma 3.4.22 (Grip–Density)**.** *Let* $a_r, b_r, b'_r, c_r, c'_r$ *be steps verifying* $b_r[\![a_r]\!]b'_r$, $c_r[\![a_r]\!]c'_r$, *and* $b'_r \ll c'_r$. *Then* $b_r \ll c_r \vee b_r \ll a_r \ll c_r$.

Proof. Let $t \xrightarrow{a_r} t'$, and say $t\,|_a = (\lambda_\theta p.s)u$, $t'\,|_{b'} = (\lambda_{\theta'} p'.s')u'$, and $t\,|_b = (\lambda_{\theta'} p''.s'')u''$; notice that the the set θ' is invariant w.r.t. the contraction of a_r. Recall that $b'_r \ll c'_r$ implies $\{p''/_{\theta'}\ u''\}$ positive, $b'12 \leqslant c'$ and $\theta' \cap \mathtt{fv}(t'\,|_{c'}) \neq \varnothing$. Observe that $\{p''/_{\theta'}\ u''\}$ positive and $\{p'/_{\theta'}\ u'\}$ decided imply $\{p'/_{\theta'}\ u'\}$ positive; cfr. Lem. 3.4.12. Let $x \in \theta' \cap \mathtt{fv}(t'|_{c'})$. Then Lem. 3.4.21 implies $x \in \mathtt{fv}(t|_c) \vee (a_r \ll c_r \wedge x \in \mathtt{fv}(u))$.

Given $b' < c'$, Lem. 3.4.18 implies $b < c$ or $(b \parallel c \wedge a2 \leqslant c)$. The latter case implies $a_r \not\ll c_r$ and $\theta' \cap \mathtt{fv}(t|_c) = \varnothing$, contradicting $x \in \mathtt{fv}(t|_c) \vee a_r \ll c_r$. Hence $b < c$. There are three cases to analyse, depending on a.

1. $a < b < c$.

   Assume $b = a12n$, $c = a12n'$ and $n < n'$, so that $b' = an$ and $c' = an'$. Then $b'12 \leqslant c'$ implies $n12 \leqslant n'$, and therefore $b12 \leqslant c$. Moreover, $a12 \leqslant b$ implies $\theta' \cap \mathtt{fv}(u) = \varnothing$, so that $x \in \mathtt{fv}(t|_c)$. Consequently, $b_r \ll c_r$.

   Assume $b = a2mn$, $c = a2m'n'$, $mn < m'n'$, $p|_m = \widehat{y}$, $p|_{m'} = \widehat{z}$, and $y, z \in \theta$. In this case, both $p|_m$ and $p|_{m'}$ being variable occurrences, along with $mn < m'n'$, imply $m = m'$, then $y = z$. Therefore $b' = akn$ and $c' = ak'n'$, where $s\,|_k = s\,|_{k'} = y$. In turn, the last assertion along $b'12 \leqslant c'$ imply $k = k'$, then $n12 \leqslant n'$, therefore $b12 \leqslant c$. Moreover, in this case $a_r \not\ll c_r$ implying $x \in \mathtt{fv}(t|_c)$. Thus $b_r \ll c_r$.

2. $b < a < c$.

   In this case $b12 = b'12 \leqslant c'$ and $a \leqslant c'$, therefore $b < a$ implies $b12 \leqslant a < c$. The existence of c'_r implies $\{p/_\theta\ u\} \neq \mathtt{fail}$. If $x \in \mathtt{fv}(t\,|_c)$, then $b_r \ll c_r$. Otherwise, $a_r \ll c_r$ and $x \in \mathtt{fv}(u) \subseteq \mathtt{fv}(t|_a)$, implying $b_r \ll a_r$. Thus we conclude.

3. $b < c < a$.

   In this case, $b12 = b'12 < c' = c$, and $a_r \not\ll c_r$ implies $x \in \mathtt{fv}(t\,|_c)$. Therefore $b_r \ll c_r$.

□

Lemma 3.4.23 (Grip–Convexity)**.** *Let* $a_r, b_r, c_r \in \mathcal{RO}(t)$ *such that* $a_r \ll b_r$ *and* $c_r < b_r$. *Then* $a_r \ll c_r \vee c_r \leqslant a_r$.

Proof. Observe that $a < b$ and $c < b$ implies that either $c \leqslant a$ or $a < c$. In the former case we immediately conclude. Otherwise, it suffices to notice that $a < c < b$, $a12 \leqslant b$ and $t|_c$ being a redex imply $a12 \leqslant c$, and that $c < b$, along with the variable convention, implies $\theta \cap \mathtt{fv}(t\,|_b) \subseteq \theta \cap \mathtt{fv}(t\,|_c)$, where $t|_a = (\lambda_\theta p.s)u$. Therefore $\varnothing \neq \theta \cap \mathtt{fv}(t|_c)$ so that we conclude $a_r \ll c_r$. □

The axioms FD and SO.

We prove the two remaining axioms.

FD is a consequence of the gripping axioms. Thm. 3.2. in [Mel96] states that an ARS satisfying the gripping axioms along with Self Reduction, Finite Residuals and Linearity, and whose embedding and gripping relations are acyclic, also enjoys FD. For the ARS modeling PPC, we have verified all the required axioms. The embedding relation being an order, and the gripping relation being included in the former, imply immediately that both are acyclic. Hence we obtain FD.

Lemma 3.4.24 (SO)**.** *Let $a_r, b_r \in \mathcal{RO}(t)$. Then there exist δ, γ such that $\delta \Vdash a_r[\![b_r]\!]$, $\gamma \Vdash b_r[\![a_r]\!]$, $\mathsf{tgt}(a_r;\gamma) = \mathsf{tgt}(b_r;\delta)$ and the relations $[\![a_r;\gamma]\!]$ and $[\![b_r;\delta]\!]$ coincide.*

Proof. We consider arbitrary reduction sequences δ, γ verifying $\delta \Vdash a_r[\![b_r]\!]$ and $\gamma \Vdash b_r[\![a_r]\!]$. Let $t \xrightarrow{a_r} t_1 \overset{\gamma}{\twoheadrightarrow} t'$, $t \xrightarrow{b_r} t_2 \overset{\delta}{\twoheadrightarrow} t''$, and $t|_a = (\lambda_\theta p.s)u$.

If $a \parallel b$, then it is straightforward to obtain $t' = t'' = t[t_1|_a]_a[t_2|_b]_b$, and also that $c_r[\![a_r;\gamma]\!] = c_r[\![b_r;\delta]\!]$ for any c_r.

If $a < b$ and $\{p/_\theta\ u\} = \mathtt{fail}$, then $\gamma = \mathtt{nil}_{t_1}$ and $t' = t_1 = t[I]_a$. On the other hand, $t_2 = t[(\lambda_\theta p'.s')u']_a$, Lem. 3.4.12 implies $\{p'/_\theta\ u'\} = \mathtt{fail}$, and $\delta = \langle\langle t_2, a\rangle\rangle$. Thus the result follows easily.

If $a < b$ and $\{p/_\theta\ u\}$ is positive, a simple, yet tedious, analysis yields the result. This analysis relies on the fact that replacement, residuals and reduction steps are compatible with substitutions. □

3.5 The normalising reduction strategy $\mathcal{S}$ for PPC

In this section, we formulate the multistep reduction strategy $\mathcal{S}$ for PPC, and we show that $\mathcal{S}$ computes necessary and non-gripping multisteps. In view of the abstract normalisation result stated in Section 3.3, these facts about $\mathcal{S}$, along with the fact that PPC as an ARS enjoys all the required axioms, imply that $\mathcal{S}$ is normalising. In the following, we use LO as an acronym for "leftmost-outermost".

We define formally the notion of **prestep**.

Definition 3.5.1. *A* ***prestep*** *is a term of the form $(\lambda_\theta\ p.t)u$, regardless of whether the match $\{p/_\theta\ u\}$ is decided or not.*

The rationale behind the definition of $\mathcal{S}$ can be described through two observations. First, it focuses on the LO prestep of t, entailing that when PPC is restricted to the λ-calculus it behaves exactly as the LO strategy for the λ-calculus. Second, if the match corresponding to the LO occurrence of a prestep is not decided, then the strategy selects only the (outermost) step, or steps, in that subterm which should be contracted to get it "closer" to a decided match.

E.g. consider the following term, where r_1 and r_2 are steps:

$$(\lambda_{\{x,y\}}\ \mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y}).y\ x)\ (\mathtt{a}\ r_1\ r_2)$$

Since we want to avoid lookahead, the strategy can only recognise the occurrences of the prestep structure, and whether a prestep is in fact a step; it cannot distinguish between different steps, in terms e.g. of their targets, or of the fact that a step could lead to an infinite reduction sequence. Therefore, the decision of $\mathcal{S}$ cannot depend on particular features of r_1 and r_2, it just knows that they are steps: it selects the steps in the same position (i.e., only r_1, only r_2, or both) for any term having this form. The match $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ r_1\ r_2\}$ is not decided; the role played by r_1 is different from that of r_2 in obtaining a decided match. Replacing r_1 by an arbitrary term t_1 does not yield a decided match, i.e. $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ t_1\ r_2\}$ is not decided. However, replacing r_2 by $\mathtt{c}\ s_2$ (resp. by $\mathtt{d}\ s_2$) does: $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ r_1\ (\mathtt{c}\ s_2)\} = \{x \to r_1, y \to s_2\}$ (resp. $\{\mathtt{a}\ \widehat{x}\ (\mathtt{c}\ \widehat{y})/_{\{x,y\}}\ \mathtt{a}\ r_1\ (\mathtt{d}\ s_2)\} = \mathtt{fail}$). Hence, contraction of r_2 can contribute towards obtaining a decided match, while contraction of r_1 does not.

Let us now consider the non-sequential example $(\lambda_{\{x\}}\mathtt{p}\,\widehat{x}\,\mathtt{m}\,\mathtt{s}.x)\ (\mathtt{p}\,(I\mathtt{a})\,(I\mathtt{f})\,(I\mathtt{d}))$. Again, $\mathcal{S}$ only knows, about the subterms $(I\mathtt{a})$, $(I\mathtt{f})$ and $(I\mathtt{d})$, that they are steps; therefore, it makes similar selections for any term having the form:

$$(\lambda_{\{x\}}\mathtt{p}\,\widehat{x}\,\mathtt{m}\,\mathtt{s}.x)\ (\mathtt{p}\ r_1\,r_2\,r_3)$$

where each r_i is a step. As in the previous example, contraction of r_1 does not contribute towards obtaining a decided match, therefore the candidates to be selected are r_2 and r_3. Selecting only r_2 could lead to non-normalisation, e.g. if r_2 is a non-terminating term, and $r_3 = I\mathtt{d}$. The situation is analogous for r_3. Hence, a normalising reduction strategy should select *both* steps in this case. This example shows the relevance of multistep reduction strategies to cope with non-sequential systems.

We also notice that steps in the *pattern* and/or the *argument* of a prestep could be selected. However, the steps in the *body* of the abstraction may be ignored, since no such step can contribute to generate a decided match. E.g., take:

$$(\lambda_{\{x,y\}}\ \mathtt{a}\ (\mathtt{b}\ \widehat{x})\ r_1.r_2)\ (\mathtt{a}\ r_3\ (\mathtt{d}\ r_4))$$

where every r_i is a step. The strategy selects r_1 and r_3; contraction of r_4 is delayed since r_1 is not in matchable form (if the contractum of r_1 were e.g. either $\mathtt{d}\ \widehat{y}$ or $\mathtt{a}$, then the match w.r.t. $\mathtt{d}\ r_4$ would be decided without the need of reducing r_4).

We model a reduction step as in Dfn. 3.4.9, i.e. a pair $\langle t, p\rangle$, where p is the position of a subterm of t having the form $(\lambda_\theta p.s)u$. The definition of the reduction strategy $\mathcal{S}$ for PPC follows.

Definition 3.5.2. *The **reduction strategy** $\mathcal{S}$ is defined as $\mathcal{S}(t) := \{\langle t, p\rangle\ /\ p \in \mathcal{S}_\pi(t)\}$, where the auxiliary functions $\mathcal{S}_\pi$ (returning positions) and $\mathcal{SM}$ (returning pairs of sets of positions) are defined as follows.*

$\mathcal{S}_\pi(x) := \varnothing$	
$\mathcal{S}_\pi(\widehat{x}) := \varnothing$	
$\mathcal{S}_\pi(\lambda_\theta\ p.t) := 1\mathcal{S}_\pi(p)$	if $p \notin \boldsymbol{NF}$
$\mathcal{S}_\pi(\lambda_\theta\ p.t) := 2\mathcal{S}_\pi(t)$	if $p \in \boldsymbol{NF}$
$\mathcal{S}_\pi((\lambda_\theta\ p.t)u) := \{\epsilon\}$	if $\{p/_\theta\ u\}$ decided
$\mathcal{S}_\pi((\lambda_\theta\ p.t)u) := 11G \cup 2D$	if $\{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle G, D\rangle \neq \langle\varnothing,\varnothing\rangle$,
$\mathcal{S}_\pi((\lambda_\theta\ p.t)u) := 11\mathcal{S}_\pi(p)$	if $\{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle\varnothing,\varnothing\rangle, p \notin \boldsymbol{NF}$
$\mathcal{S}_\pi((\lambda_\theta\ p.t)u) := 12\mathcal{S}_\pi(t)$	if $\{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle\varnothing,\varnothing\rangle, p \in \boldsymbol{NF}, t \notin \boldsymbol{NF}$
$\mathcal{S}_\pi((\lambda_\theta\ p.t)u) := 2\mathcal{S}_\pi(u)$	if $\{p/_\theta\ u\} = \mathtt{wait}, \mathcal{SM}_\theta(p,u) = \langle\varnothing,\varnothing\rangle, p \in \boldsymbol{NF}, t \in \boldsymbol{NF}$
$\mathcal{S}_\pi(tu) := 1\mathcal{S}_\pi(t)$	if t *is not an abstraction and* $t \notin \boldsymbol{NF}$
$\mathcal{S}_\pi(tu) := 2\mathcal{S}_\pi(u)$	if t *is not an abstraction and* $t \in \boldsymbol{NF}$

$$
\begin{array}{rll}
\mathcal{SM}_\theta(\hat{x}, t) := & \langle \varnothing, \varnothing \rangle & \text{if } x \in \theta \\
\mathcal{SM}_\theta(\hat{x}, \hat{x}) := & \langle \varnothing, \varnothing \rangle & \text{if } x \notin \theta \\
\mathcal{SM}_\theta(p_1 p_2, t_1 t_2) := & \langle 1G_1 \cup 2G_2, 1D_1 \cup 2D_2 \rangle & \text{if } t_1 t_2, p_1 p_2 \in \boldsymbol{MF}, \mathcal{SM}_\theta(p_i, t_i) = \langle G_i, D_i \rangle \\
\mathcal{SM}_\theta(p, t) := & \langle \mathcal{S}_\pi(p), \varnothing \rangle & \text{if } p \notin \boldsymbol{MF} \\
\mathcal{SM}_\theta(p, t) := & \langle \varnothing, \mathcal{S}_\pi(t) \rangle & \text{if } p \in \boldsymbol{MF} \ \& \ t \notin \boldsymbol{MF} \ \& \ \neg bm(p, \theta)
\end{array}
$$

Recall Dfn. 2.1.8 for the definition of the set of *normal forms*, i.e. $\boldsymbol{NF}$.

The function $\mathcal{SM}$ formalises the simultaneous structural analysis of pattern and argument which is performed if the LO prestep is not decided. Its outcome is a pair of sets of positions, corresponding to steps inside the pattern and argument respectively, which could contribute to turning a non-decided match into a decided one. Notice the similarities between the first three clauses in the definition of $\mathcal{SM}$ and those of the definition of the matching operation (cfr. Section 3.4.1).

We also notice that whenever a non-decided match can be turned into a decided one, the function $\mathcal{SM}$ chooses at least one (contributing) step. Formally, it can be proved that, given p and u such that $\{p/_\theta\ u\} = \texttt{wait}$, if there exist p' and u' such that $p \twoheadrightarrow p'$, $u \twoheadrightarrow u'$ and $\{p'/_\theta\ u'\}$ is decided, then $\mathcal{SM}_\theta(p, u) \neq \langle \varnothing, \varnothing \rangle$.

Let us analyse briefly the clauses in the definition of $\mathcal{S}_\pi$.

The focus on the LO prestep of a term is formalised in the first four and the last two clauses. If the LO prestep is in fact a step, then the strategy selects exactly that step; this is the meaning of the fifth clause. If the LO prestep is not a step, then $\mathcal{SM}$ is used. If it returns some steps which could contribute towards a decided match, then the strategy selects them (sixth clause). Otherwise, as we already remarked, the prestep will never turn into a step, so that the strategy looks for the LO prestep inside the components of the term (seventh, eighth and ninth clauses).

Notice that for the translation of any λ-calculus term into PPC, all the presteps are in fact steps, particularly the LO one. Therefore the focus on the LO prestep, along with the selection of exactly that prestep if it is a step, imply that $\mathcal{S}$ behaves exactly as LO when PPC is restricted to the λ-calculus.

While the strategy focuses in obtaining a decided match for the LO prestep, it can select more steps than needed. E.g., for the term $(\lambda_{\{y\}}\ \texttt{a}\ \texttt{b}\ \texttt{c}\ \hat{y}.y)\ (\texttt{a}\ (I\ \texttt{c})\ (I\ \texttt{b})\ (I\ \texttt{a}))$, the set selected by the strategy $\mathcal{S}$ is $\{I\ \texttt{c}, I\ \texttt{b}\}$, even if the contraction of just one step of the set suffices to make the head match decided.

The reduction strategy $\mathcal{S}$ is complete, i.e., if t is not a normal form, then $\mathcal{S}(t) \neq \varnothing$. Moreover, all steps in $\mathcal{S}(t)$ are *outermost*. On the other hand, notice that $\mathcal{S}$ is not *outermost fair* [vR97]. Indeed, given $(\lambda\, \texttt{c}\, x.s)\, \Omega$, where Ω is as in the λ-calculus, $\mathcal{S}$ continuously contracts Ω, even when s contains a step.

3.5.1 Normalisation of $\mathcal{S}$ – preliminary notions and results

The remainder of this section is devoted to prove that $\mathcal{S}$ selects necessary and non-gripping sets. We use the notational conventions a_r and A_r, cfr. Notation 3.4.10 in page 74, to relate (sets of) steps with the corresponding (sets of) positions.

The forthcoming proofs rely on the notion of *projection* of a multireduction w.r.t. a position. We define this notion in the following. Afterwards, we prove that it is well defined, and that moreover target, residuals and the *uses* relation are compatible with projections. Several previous definitions are needed.

Notation 3.5.3. *Let $\mathcal{B} \subseteq \mathcal{RO}(t)$ and $a \in \texttt{Pos}(t)$. We write $a \leqslant \mathcal{B}$ iff $a \leqslant b$ for all $b_r \in \mathcal{B}$. Analogously, for every reduction sequence δ and $a \in \texttt{Pos}(src(\delta))$, we write $a \leqslant \delta$ iff for any $i \leqslant |\delta|$, $a \leqslant b_i$ where $\delta[i] = \langle t_i, b_i \rangle$.*

Definition 3.5.4. *Let $\mathcal{B}$ be a multistep, and $a \in \texttt{Pos}(src(\mathcal{B}))$. We say that $\mathcal{B}$ **preserves** a iff all $b_r \in \mathcal{B}$ verify $b \nless a$, or equivalently, $a \leqslant b$ or $a \parallel b$. In turn, a* multireduction Δ ***preserves*** *a iff all its elements do.*

Definition 3.5.5. *If $\mathcal{B}$ preserves a, then we define the **free part** and the **dominated part** of $\mathcal{B}$ w.r.t. a, written $\mathcal{B}^F_a$ and $\mathcal{B}^D_a$ respectively, as follows: $\mathcal{B}^F_a := \{b_r \in \mathcal{B} \mathrel{/} a \parallel b\}$ and $\mathcal{B}^D_a := \{b_r \in \mathcal{B} \mathrel{/} a \leqslant b\}$.[7] Observe $\mathcal{B} = \mathcal{B}^F_a \uplus \mathcal{B}^D_a$, and $b_{1r} \in \mathcal{B}^F_a$ and $b_{2r} \in \mathcal{B}^D_a$ imply $b_1 \parallel b_2$.*

Definition 3.5.6. *Let δ be a reduction sequence, and $a \in \texttt{Pos}(t)$ where $t = src(\delta)$. We define the **projection** of δ w.r.t. a, notation $\delta|_a$, as follows: if $\delta = \texttt{nil}_t$, then $\delta|_a = \texttt{nil}_{t|_a}$, otherwise $|\delta|_a| = |\delta|$ and $\delta|_a[i] = \delta[i]|_a$ for all $i \leqslant |\delta|$.*

Definition 3.5.7. *If $\mathcal{B} \subseteq \mathcal{RO}(t)$ preserves $a \in \texttt{Pos}(t)$, then we define the **projection of $\mathcal{B}$ w.r.t.** a, notation $\mathcal{B}|_a$, as $\{\langle t|_a, b' \rangle \mathrel{/} ab'_r \in \mathcal{B}\}$; if this set is empty, then $\mathcal{B}|_a = \varnothing_{t|_a}$. Notice that $\mathcal{B}|_a = \mathcal{B}^D_a|_a$.*

Definition 3.5.8. *If a multireduction Δ preserves $a \in \texttt{Pos}(src(\Delta))$, then we define **projection of Δ w.r.t.** a, notation $\Delta|_a$, as follows: $\texttt{nil}_t|_a = \texttt{nil}_{t|_a}$, and in any other case, $\Delta|_a = \langle \Delta[1]|_a; \ldots; \Delta[n]|_a; \ldots \rangle$.*

We prove that $\delta|_a$ is a well-defined reduction sequence (Lem. 3.5.9, along with a straightforward induction on $|\delta|$, suffices), and that targets (Lem. 3.5.10) and residuals (Lem. 3.5.12) are compatible with the projection of reduction sequences.

Lemma 3.5.9. *Let $t \xrightarrow{ab_r} t'$. Then $t|_a \xrightarrow{b_r} t'|_a$.*

Proof. Let $t|_{ab} = (\lambda_\theta p.s)u$ and $s' = \{p/_\theta\ u\}s$. Then $t' = t[s']_{ab}$. Observe $(t|_a)|_b = t|_{ab}$ and $t' = t[(t|_a)[s']_b]_a$ implying $t'|_a = (t|_a)[s']_b$. Thus we conclude. □

Lemma 3.5.10. *Let a be a position and $t \overset{\delta}{\twoheadrightarrow} t'$, such that $a \leqslant \delta$. Then $t|_a \overset{\delta|_a}{\twoheadrightarrow} t'|_a$.*

Proof. We proceed by induction on $|\delta|$. If $\delta = \texttt{nil}_t$, then $t' = t$ and $\delta|_a = \texttt{nil}_{t|_a}$, so we conclude. Otherwise, $a \leqslant \delta$ implies $\delta = ab_r; \delta'$, say $t \xrightarrow{ab_r} t'' \overset{\delta'}{\twoheadrightarrow} t'$. Then Lem. 3.5.9 and IH imply $t|_a \xrightarrow{b_r} t''|_a \overset{\delta'|_a}{\twoheadrightarrow} t'|_a$. Thus we conclude. □

Lemma 3.5.11. *Let $ab_r, ac_r \in \mathcal{RO}(t)$, so that $b_r, c_r \in \mathcal{RO}(t|_a)$. Then $ac_r[\![ab_r]\!]d_r$ iff $d = ad'$ and $c_r[\![b_r]\!]d'_r$.*

Proof. Let $t|_{ab} = (t|_a)|_b = (\lambda_\theta p.s)u$. In the analysis of $ac_r[\![ab_r]\!]d_r$ and $c_r[\![b_r]\!]d'_r$, cfr. Dfn. 3.4.9, always the case applying is the same, and moreover with the same arguments. E.g. if $ab = ac2mn$, then $b = c2mn$, the values for m and n coincide. In this case, the subterms p and s also coincide. These observations suffice to conclude. □

---

[7] A remark about the names "free" and "dominated" given to $\mathcal{B}^F_a$ and $\mathcal{B}^D_a$ follows. We recall that b is free from a (that is, $b \not\sqsupseteq a$) iff $a \nleqslant b$, i.e. $b < a$ or $b \parallel a$. The former possibility cannot occur since $\mathcal{B}$ preserves a, hence the name given to $\mathcal{B}^F_a$. In turn, it is not true in general that $b \in \mathcal{B}^D_a$ implies that b is dominated by $\{a\}$, the exception being the case $b = a$; hence, the name "dominated" is in fact approximate.

Lemma 3.5.12. *Let a be a position, $ab_r \in \mathcal{RO}(t)$, so that $b_r \in \mathcal{RO}(t|_a)$, and δ a reduction sequence verifying $\mathsf{src}(\delta) = t$ and $a \leqslant \delta$. Then $ab_r[\![\delta]\!]d_r$ iff $d = ad'$ and $b_r[\![\delta|_a]\!]d'_r$.*

Proof. We proceed by induction on $|\delta|$. If $\delta = \mathtt{nil}_t$, so that $\delta|_a = \mathtt{nil}_{t|_a}$, then $ab_r[\![\delta]\!]d_r$ implies $d = ab$, and $b_r[\![\delta|_a]\!]d'_r$ implies $d' = b$, thus we conclude. Otherwise, $a \leqslant \delta$ implies $\delta = ac_r; \delta'$, $a \leqslant \delta'$, and $\delta|_a = c_r; \delta'|_a$. We proceed by double implication. Let us define $t' = \mathsf{src}(\delta')$.
$\Longrightarrow$) $ab_r[\![\delta]\!]d_r$ implies $ab_r[\![ac_r]\!]e_r$ and $e_r[\![\delta']\!]d_r$ for some e_r. Lem. 3.5.11 implies $e = ae'$ and $b_r[\![c_r]\!]e'_r$. Observe that $e_r = ae'_r \in \mathcal{RO}(t')$. Therefore IH yields $d = ad'$ and $e'_r[\![\delta'|_a]\!]d'_r$, hence $b_r[\![\delta|_a]\!]d'_r$.
$\Longleftarrow$) $b_r[\![\delta|_a]\!]d'_r$ implies $b_r[\![c_r]\!]e'_r$ and $e'_r[\![\delta'|_a]\!]d'_r$ for some e'_r. Let us call $e = ae'$ and $d = ad'$. Observe $e'_r \in \mathcal{RO}(t'|_a)$, cfr. Lem. 3.5.9, then $e_r \in \mathcal{RO}(t')$. Lem. 3.5.11 implies $ab_r[\![ac_r]\!]e_r$. In turn, IH implies $e_r[\![\delta']\!]d_r$. Thus we conclude. □

We verify that if $a \leqslant \mathcal{B}$, then residuals (Lem. 3.5.15) and complete developments (Lem. 3.5.16) are compatible with the projection $\mathcal{B}|_a$.

Lemma 3.5.13. *Let $a \leqslant \mathcal{B}$ and $b_r \in \mathcal{B}$. Then $a \leqslant \mathcal{B}[\![b_r]\!]$.*

Proof. Hypotheses imply $b = ab'$. For all $c_r \in \mathcal{B}[\![ab'_r]\!]$, Lem. 3.5.11 implies $c = ac'$. Thus we conclude. □

Lemma 3.5.14. *Let $a \leqslant \mathcal{B}$ and $\delta \Vdash \mathcal{B}$. Then $a \leqslant \delta$.*

Proof. We proceed by induction on $\nu(\mathcal{B})$. Let $t \xrightarrow{\mathcal{B}} t'$. If $\mathcal{B} = \varnothing_t$ then $\delta = \mathtt{nil}_t$ and we conclude immediately. Otherwise $\mathcal{B} = b_r; \delta'$ where $b_r \in \mathcal{B}$, implying $a \leqslant b$, and $\delta' \Vdash \mathcal{B}[\![b_r]\!]$. Lem. 3.5.13 implies $a \leqslant \mathcal{B}[\![b_r]\!]$. Hence IH yields $a \leqslant \delta'$, which suffices to conclude. □

Lemma 3.5.15. *Let $a \leqslant \mathcal{B}$ and $ab_r \in \mathcal{B}$. Then $(\mathcal{B}[\![ab_r]\!])|_a = \mathcal{B}|_a[\![b_r]\!]$.*

Proof. By double inclusion.
$\supseteq$) Let $c_r \in (\mathcal{B}[\![ab_r]\!])|_a$, so that $ac_r \in \mathcal{B}[\![ab_r]\!]$. Let $ad_r \in \mathcal{B}$ such that $ad_r[\![ab_r]\!]ac_r$, observe $d_r \in \mathcal{B}|_a$. Lem. 3.5.11 implies $d_r[\![b_r]\!]c_r$. Hence $c_r \in \mathcal{B}|_a[\![b_r]\!]$.
$\subseteq$) Let $c_r \in \mathcal{B}|_a[\![b_r]\!]$, let $d_r \in \mathcal{B}|_a$ such that $d_r[\![b_r]\!]c_r$, observe that $ad_r \in \mathcal{B}$. Lem. 3.5.11 implies $ad_r[\![ab_r]\!]ac_r$. Then $ac_r \in \mathcal{B}[\![ab_r]\!]$, implying $c_r \in (\mathcal{B}[\![ab_r]\!])|_a$. □

Lemma 3.5.16. *Let $a \leqslant \mathcal{B}$ and $\delta \Vdash \mathcal{B}$. Then $\delta|_a \Vdash \mathcal{B}|_a$.*

Proof. By induction on $\nu(\mathcal{B})$. Let $t = \mathsf{src}(\mathcal{B})$. If $\mathcal{B} = \varnothing_t$ then observing $\delta = \mathtt{nil}_t$ suffices to conclude. Otherwise $\delta = ab_r; \delta'$ where $\delta' \Vdash \mathcal{B}[\![ab_r]\!]$. In this case, $\delta|_a = b_r; \delta'|_a$. IH yields $\delta'|_a \Vdash (\mathcal{B}[\![ab_r]\!])|_a$. In turn, Lem. 3.5.15 implies $(\mathcal{B}[\![ab_r]\!])|_a = \mathcal{B}|_a[\![b_r]\!]$. Hence $\delta|_a \Vdash \mathcal{B}|_a$. □

We verify that given a multistep $t \xrightarrow{\mathcal{B}} t'$ s.t. $\mathcal{B}$ preserves a, it is only the dominated part of $\mathcal{B}$ that actually modifies $t|_a$; cfr. Lem. 3.5.18.

Lemma 3.5.17. *Let $a, \mathcal{B}$ such that $\mathcal{B}$ preserves a, and $b_r \in \mathcal{B}$. Then $\mathcal{B}[\![b_r]\!]$ preserves a. Moreover $\mathcal{B}[\![b_r]\!]^F_a = \mathcal{B}^F_a[\![b_r]\!]$ and $\mathcal{B}[\![b_r]\!]^D_a = \mathcal{B}^D_a[\![b_r]\!]$.*

Proof. Take $b'_{1r} \in \mathcal{B}[\![b_r]\!]$ and let $b_{1r} \in \mathcal{B}$ such that $b_{1r}[\![b_r]\!]b'_{1r}$. Observe that either $b \leqslant b'_1$ (if $b < b_1$), or $b'_1 = b_1$ (if $b \nleqslant b_1$). We verify that $b'_1 \nless a$. $\mathcal{B}$ preserves a implies $a \leqslant b$ or $a \parallel b$, and analogously for b_1.

- Assume $a \leqslant b$. If $a \parallel b_1$ then $b'_1 = b_1$ implying $a \parallel b'_1$. If $a \leqslant b_1$, then either $b'_1 = b_1$ or $b \leqslant b'_1$ imply $a \leqslant b'_1$.
- Assume $a \parallel b$. If $a \parallel b_1$ then either $b'_1 = b_1$ or $b \leqslant b'_1$ imply $a \parallel b'_1$. If $a \leqslant b_1$, so that $b \parallel b_1$, then $b'_1 = b_1$, implying $a \leqslant b'_1$.

Consequently, $\mathcal{B}[\![b_r]\!]$ preserves a. Furthermore, $a \parallel b_1$ implies $a \parallel b'_1$ and $a \leqslant b_1$ implies $a \leqslant b'_1$. The former assertion implies $\mathcal{B}^F_a[\![b_r]\!] \subseteq \mathcal{B}[\![b_r]\!]^F_a$. Moreover, let $b'_{2r} \in \mathcal{B}[\![b_r]\!]^F_a$ and $b_{2r} \in \mathcal{B}$ such that $b_{2r}[\![b_r]\!]b'_{2r}$. Observe that $a \leqslant b_2$ would imply $a \leqslant b'_2$, therefore $\mathcal{B}$ preserves a implies $a \parallel b_2$, i.e. $b_{2r} \in \mathcal{B}^F_a$. Therefore $\mathcal{B}[\![b_r]\!]^F_a \subseteq \mathcal{B}^F_a[\![b_r]\!]$, so that we obtain $\mathcal{B}[\![b_r]\!]^F_a = \mathcal{B}^F_a[\![b_r]\!]$. An analogous argument on the dominated parts allows to conclude. □

Lemma 3.5.18. *Let $\mathcal{B} \in \mathcal{RO}(t)$ and assume $\mathcal{B}$ preserves a and $t \xrightarrow{\mathcal{B}^D_a} t'' \xrightarrow{\mathcal{B}^F_a[\![\mathcal{B}^D_a]\!]} t'$. Then $t'|_a = t''|_a$.*

Proof. A simple induction based on Lem. 3.5.17 yields that $b \parallel a$ if $b_r \in \mathcal{B}^F_a[\![\mathcal{B}^D_a]\!]$. Therefore, a straightforward analysis allows to conclude. □

Lem. 3.5.18 allows to verify that targets and residuals are compatible with the projection $\mathcal{B}\,|_a$.

Lemma 3.5.19. *Let $t \xrightarrow{\mathcal{B}} t'$ and assume $\mathcal{B}$ preserves a. Then:*

(i) $t|_a \xrightarrow{\mathcal{B}|_a} t'|_a$.

(ii) If $ac_r \in \mathcal{RO}(t)$, so that $c_r \in \mathcal{RO}(t|_a)$, then $ac_r[\![\mathcal{B}]\!]d_r$ iff $d = ad'$ and $c_r[\![\mathcal{B}|_a]\!]d'_r$.

Proof. Let $t \xrightarrow{\mathcal{B}^D_a} t'' \xrightarrow{\mathcal{B}^F_a[\![\mathcal{B}^D_a]\!]} t'$. Let δ such that $\delta \Vdash \mathcal{B}^D_a$, and $\gamma \Vdash \mathcal{B}^F_a[\![\mathcal{B}^D_a]\!]$. Observe $t \xrightarrow{\delta} t'' \xrightarrow{\gamma} t'$. Moreover, $a \leqslant \delta$ and $\delta|_a \Vdash \mathcal{B}^D_a|_a = \mathcal{B}|_a$, by Lem. 3.5.14 and Lem. 3.5.16 respectively. On the other hand, $b \parallel a$ for all $b_r \in \mathcal{B}^F_a[\![\mathcal{B}^D_a]\!]$ implies $a \parallel \gamma[i]$ for all i. Notice that $a \parallel b \wedge a \parallel c$ implies $a \parallel d$ whenever $b_r[\![c_r]\!]d_r$.

To prove item (i), it suffices to observe that Lem. 3.5.10 implies $t\,|_a \xrightarrow{\delta|_a} t''\,|_a = t'\,|_a$; cfr. Lem. 3.5.18.

We prove item (ii), by double implication.

$\Longrightarrow$) Let $ac_r[\![\mathcal{B}]\!]d_r$. Then $ac_r[\![\delta]\!]e_r$ and $e_r[\![\gamma]\!]d_r$ for some e_r. Lem. 3.5.12 implies $e = ae'$ and $c_r[\![\delta\,|_a]\!]e'_r$. In turn, $a \parallel \gamma[i]$ for all i and $a \leqslant e$ imply $d = e$, i.e. $d = ad'$ where $d' = e'$, and $c_r[\![\delta|_a]\!]d'_r$. We conclude by recalling that $\delta|_a \Vdash \mathcal{B}|_a$.

$\Longleftarrow$) Let $c_r[\![\mathcal{B}|_a]\!]d'_r$, and $d = ad'$. Then $c_r[\![\delta|_a]\!]d'_r$. Lem. 3.5.12 implies $ac_r[\![\delta]\!]\langle t'', d\rangle$. In turn, $a \parallel \gamma[i]$ for all i and $a \leqslant d$ imply $\langle t'', d\rangle[\![\gamma]\!]\langle t', d\rangle$. Hence $ac_r[\![\mathcal{B}]\!]d_r$ □

Now consider a *multireduction* Δ which preserves some position a. For any $n < |\Delta|$, Lem. 3.5.19:(i) implies that $\mathsf{src}(\Delta[n+1]\,|_a) = \mathsf{src}(\Delta[n+1])\,|_a = \mathsf{tgt}(\Delta[n]\,|_a)$. This implies that the definition of the projection of Δ over a is well-defined.

We verify the expected properties for the projections of multireductions.

Lemma 3.5.20. *Let $t \xrightarrow{\Delta} t'$ and assume Δ preserves a. Then:*

(i) $t|_a \xrightarrow{\Delta|_a} t'|_a$.

(ii) If $ac_r \in \mathcal{RO}(t)$, then $ac_r[\![\Delta]\!]d_r$ iff $d = ad'$ and $c_r[\![\Delta|_a]\!]d'_r$.

(iii) If $ac_r \in \mathcal{RO}(t)$, then Δ uses ac_r iff $\Delta|_a$ uses c_r.

Proof. To prove item (i) a simple induction on $|\Delta|$, resorting on Lem. 3.5.19:(i), suffices.

Item (ii) admits an argument similar to the one used to prove Lem. 3.5.12, resorting on Lem. 3.5.19:(i) instead of Lem. 3.5.11.

We prove item (iii). Assume $\Delta|_a$ uses c_r, i.e. $\Delta = \Delta_1; \mathcal{D}; \Delta_2$ and there exists some $d_r \in \mathcal{D}|_a \cap c_r[\![\Delta_1|_a]\!]$. Item (ii) implies $ac_r[\![\Delta_1]\!]ad_r$, and moreover $d_r \in \mathcal{D}|_a$ implies $ad_r \in \mathcal{D}$. Hence Δ uses ac_r.

Assume Δ uses ac_r, i.e. $\Delta = \Delta_1; \mathcal{D}; \Delta_2$ and there exists some $d_r \in \mathcal{D} \cap ac_r[\![\Delta_1]\!]$. Item (ii) implies $d = ad'$, so that $d'_r \in \mathcal{D}|_a$, and $c_r[\![\Delta_1|_a]\!]d'_r$. On the other hand, $\Delta|_a = \Delta_1|_a; \mathcal{D}|_a; \Delta_2|_a$. Hence $\Delta|_a$ uses c_r. □

We conclude this section by introducing some results used to prove that $\mathcal{S}$ selects necessary sets, and also to prove that $\mathcal{S}$ selects non-gripping sets. In the following proofs, as well as in those of Section 3.5.2, we use Notation 3.4.10 in the following sense: if $\mathcal{SM}_\theta(p,t) = \langle G, D\rangle$, then G_r and D_r are the sets of steps, in p and t respectively, corresponding to the sets of positions G and D.

Lemma 3.5.21. *If $\{\!\{p \rhd_\theta u\}\!\}$ is positive, then $\mathcal{SM}_\theta(p,u) = \langle\varnothing,\varnothing\rangle$.*

Proof. Observe that $\{\!\{p \rhd_\theta u\}\!\}$ positive implies $p \in \boldsymbol{DS}$. Then a simple induction on p suffices. In particular, if $p = p_1p_2$, then $\{\!\{p \rhd_\theta u\}\!\}$ positive implies $u = u_1u_2$ and both $\{\!\{p_i \rhd_\theta u_i\}\!\}$ positive, so that the IH on each i allows to conclude. □

Lemma 3.5.22. *Let t, u be terms and p be a pattern.*

(i) Let $t \xrightarrow{\Delta} t'$ where $t \notin \boldsymbol{MF}$ and $t' \in \boldsymbol{MF}$. Then Δ uses $\mathcal{S}(t)$ and $\mathcal{S}(t)[\![\Delta]\!] = \varnothing$.

(ii) Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$, where $\{\!\{p \rhd_\theta u\}\!\} = \texttt{wait}$ and $\{\!\{p' \rhd_\theta u'\}\!\}$ is decided. Let $\langle G, D\rangle = \mathcal{SM}_\theta(p,u)$. Then Γ uses G_r or Π uses D_r. Moreover, $\{\!\{p' \rhd_\theta u'\}\!\}$ positive implies $G_r[\![\Gamma]\!] = D_r[\![\Pi]\!] = \varnothing$.

(iii) Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$, where $\{p/_\theta\ u\} = \texttt{wait}$ and $\{p'/_\theta\ u'\}$ is decided. Let $\langle G, D\rangle = \mathcal{SM}_\theta(p,u)$. Then Γ uses G_r or Π uses D_r. Moreover, $\{p'/_\theta\ u'\}$ positive implies $G_r[\![\Gamma]\!] = D_r[\![\Pi]\!] = \varnothing$.

Proof. Item (iii) follows from item (ii) since $\{p/_\theta\ u\} = \texttt{wait}$ implies $\{\!\{p \rhd_\theta u\}\!\} = \texttt{wait}$, and $\{p'/_\theta\ u'\}$ decided or positive implies $\{\!\{p' \rhd_\theta u'\}\!\}$ decided and positive respectively. We prove items (i) and (ii), by simultaneous induction on $|t| + |u| + |p|$.

- Item (i). Observe that $t \notin \boldsymbol{MF}$ implies that t is either a variable or an application. In the former case $t' = t \notin \boldsymbol{MF}$ contradicting the hypothesis. So we consider the latter one.

  Assume $t = (\lambda_\theta p.s)u$ where $\{p/_\theta\ u\}$ is decided, so that $\mathcal{S}(t) = \{\langle t, \epsilon\rangle\}$. If there is some $i \leqslant |\Delta|$ such that $\langle t_i, \epsilon\rangle \in \Delta[i]$, where $t_i \xrightarrow{\Delta[i]} t_{i+1}$, taking the minimal such i yields $\mathcal{S}(t)[\![\Delta[1..i-1]]\!] = \{\langle t_i, \epsilon\rangle\}$, so that Δ uses $\mathcal{S}(t)$, and moreover

$\mathcal{S}(t)[\![\Delta[1..i]]\!] = \epsilon$. Otherwise $t' = (\lambda_\theta p'.s')u'$, contradicting $t' \in \boldsymbol{MF}$. Thus we conclude.

Assume $t = (\lambda_\theta p.s)u$ where $\{p/_\theta\ u\} = \mathtt{wait}$. Then $t' \in \boldsymbol{MF}$ implies $t \xrightarrow{\Delta'} t'' \xrightarrow{\Delta''} t'$ where $t'' = (\lambda_\theta p''.s'')u''$ and $\{p''/_\theta\ u''\}$ is decided. Moreover Δ' preserves 11 and 2, implying $p \xrightarrow{\Delta'|_{11}} p''$ and $u \xrightarrow{\Delta'|_2} u''$ by Lem. 3.5.20:(i). Let $\mathcal{SM}_\theta(p, u) = \langle G, D\rangle$. The IH:(iii) can be applied, yielding that $\Delta'|_{11}$ uses G_r or $\Delta'|_2$ uses D_r. Therefore $\langle G, D\rangle \neq \langle\varnothing, \varnothing\rangle$, implying $\mathcal{S}_\pi(t) = 11G \cup 2D$. In turn, Lem. 3.5.20:(iii) implies that Δ' uses $\mathcal{S}(t)$. Furthermore, if $\{p''/_\theta\ u''\}$ is positive, then IH:(iii) also implies $G_r[\![\Delta'|_{11}]\!] = D_r[\![\Delta'|_2]\!] = \varnothing$, and $\{p''/_\theta\ u''\} = \mathtt{fail}$, along with $t' \in \boldsymbol{MF}$, implies $t' = I$. In both cases we obtain $\mathcal{S}(t)[\![\Delta]\!] = \varnothing$.

Assume $t = su$ where $s \notin \boldsymbol{MF}$. Then, $t' \in \boldsymbol{MF}$ implies $t = su \xrightarrow{\Delta'} s'u' \xrightarrow{\Delta''} t'$, where Δ' preserves 1 and 2, and either $s' \in \boldsymbol{DS}$ or s' is an abstraction, i.e. $s' \in \boldsymbol{MF}$. In turn, Lem. 3.5.20:(i) implies $s \xrightarrow{\Delta'|_1} s'$. Therefore, the IH:(i) applies, yielding that $\Delta'|_1$ uses $\mathcal{S}(s)$ and $\mathcal{S}(s)[\![\Delta'|_1]\!] = \varnothing$. Observe that $s \notin \boldsymbol{MF}$ and $s' \in \boldsymbol{MF}$ imply $s \neq s'$, then $s \notin \boldsymbol{NF}$, hence $\mathcal{S}_\pi(t) = 1\mathcal{S}_\pi(s)$. Hence Lem. 3.5.20:(iii) and Lem. 3.5.20:(ii) implies that Δ' uses $\mathcal{S}(t)$ and $\mathcal{S}(t)[\![\Delta']\!] = \varnothing$ respectively. Thus we conclude.

Finally, the remaining case $t = su$ where $s \in \boldsymbol{DS}$ contradicts $t \notin \boldsymbol{MF}$.

- Item (ii). Observe that $\{\!\{p' \rhd_\theta u'\}\!\}$ decided implies $p' \in \boldsymbol{MF}$, and also $u' \in \boldsymbol{MF}$ unless $bm(p', \theta)$. We consider the following cases depending on whether p is in $\boldsymbol{MF}$ or not and likewise for u.

  Assume $p \notin \boldsymbol{MF}$, so that $G = \mathcal{S}_\pi(p)$ and $D = \varnothing$. In this case, $p' \in \boldsymbol{MF}$ implies that the IH:(i) can be applied on $p \xrightarrow{\Gamma} p'$. We obtain that Γ uses G_r and $G_r[\![\Gamma]\!] = \varnothing$, which suffices to conclude.

  Assume $p \in \boldsymbol{MF}$ and $u \notin \boldsymbol{MF}$, so that $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{wait}$ implies $\neg bm(p, \theta)$, and therefore $G = \varnothing$ and $D = \mathcal{S}_\pi(u)$. Observe that $p \in \boldsymbol{MF}$, $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{wait}$ and $p \xrightarrow{\Gamma} p'$ imply $\neg bm(p', \theta)$, so that $u' \in \boldsymbol{MF}$. Therefore, the IH:(i) can be applied on $u \xrightarrow{\Pi} u'$. We conclude like in the previous case.

  Assume $p, u \in \boldsymbol{MF}$, so that $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{wait}$ implies $p = p_1p_2$ and $u = u_1u_2$. Then $G = 1G_1 \cup 2G_2$ and $D = 1D_1 \cup 2D_2$, where $\mathcal{SM}_\theta(p_i, u_i) = \langle G_i, D_i\rangle$ for $i = 1, 2$. Moreover, it is straightforward to verify that both Γ and Π preserve 1 and 2, so that Lem. 3.5.20:(i) implies $p' = p'_1p'_2$, $u' = u'_1u'_2$, and $p_i \xrightarrow{\Gamma|_i} p'_i$ and $u_i \xrightarrow{\Pi|_i} u'_i$ for $i = 1, 2$. On the other hand, the hypotheses imply the existence of some $k \in \{1, 2\}$ verifying $\{\!\{p_k \rhd_\theta u_k\}\!\} = \mathtt{wait}$ and $\{\!\{p'_k \rhd_\theta u'_k\}\!\}$ decided. Therefore, the IH:(ii) can be applied yielding that $\Gamma|_k$ uses $(G_k)_r$ or $\Pi|_k$ uses $(D_k)_r$. Hence, Lem. 3.5.20:(iii) implies that Γ uses G_r or Π uses D_r.

  Moreover, $\{\!\{p' \rhd_\theta u'\}\!\}$ positive implies $\{\!\{p'_i \rhd_\theta u'_i\}\!\}$ positive for $i = 1, 2$. For each i, observe that $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{wait}$ implies $\{\!\{p_i \rhd_\theta u_i\}\!\} \neq \mathtt{fail}$. If $\{\!\{p_i \rhd_\theta u_i\}\!\} = \mathtt{wait}$, then the IH:(ii) implies $(G_i)_r[\![\Gamma|_i]\!] = (D_i)_r[\![\Pi|_i]\!] = \varnothing$; if $\{\!\{p_i \rhd_\theta u_i\}\!\}$ is positive, then Lem. 3.5.21 implies $G_i = D_i = \varnothing$. Hence Lem. 3.5.20:(ii) yields $G_r[\![\Gamma]\!] = D_r[\![\Pi]\!] = \varnothing$.

□

3.5.2 Normalisation of $\mathcal{S}$ – main proofs

In this section we show that $\mathcal{S}$ always selects *necessary* and *non-gripping* sets of redexes.

Proposition 3.5.23. *Let $t \xrightarrow{\Delta}\!\!\!\!\rightarrow t'$ where $t \notin \boldsymbol{NF}$ and $t' \in \boldsymbol{NF}$. Then Δ uses $\mathcal{S}(t)$.*

Proof. We prove the following three statements simultaneously, where t, u, p are terms.

(i) The statement of the proposition.

(ii) Let $p \xrightarrow{\Gamma}\!\!\!\!\rightarrow p'$ and $u \xrightarrow{\Pi}\!\!\!\!\rightarrow u'$ where $p', u' \in \boldsymbol{NF}$, $\langle G, D\rangle = \mathcal{SM}_\theta(p, u) \neq \langle \varnothing, \varnothing\rangle$, and $\{\!\{p \rhd_\theta u\}\!\} = \{\!\{p' \rhd_\theta u'\}\!\} = \texttt{wait}$. Then Γ uses G_r or Π uses D_r.

(iii) Let $p \xrightarrow{\Gamma}\!\!\!\!\rightarrow p'$ and $u \xrightarrow{\Pi}\!\!\!\!\rightarrow u'$ where $p', u' \in \boldsymbol{NF}$, $\langle G, D\rangle = \mathcal{SM}_\theta(p, u) \neq \langle \varnothing, \varnothing\rangle$, and $\{p/_\theta\ u\} = \{p'/_\theta\ u'\} = \texttt{wait}$. Then Γ uses G_r, or Π uses D_r.

As in Lem. 3.5.22, item (iii) follows from item (ii). So we prove the others, by induction on $|t| + |u| + |p|$.

- Item (i). If t is either a matchable or a variable, then t is a normal form contradicting the hypotheses, so let us assume that t is an application or an abstraction.

  Assume $t = (\lambda_\theta p.s)u$ and $\{p/_\theta\ u\}$ decided, so that $\mathcal{S}(t) = \{\langle t, \epsilon\rangle\}$. Suppose Δ does not use $\mathcal{S}(t)$, so that $t' = (\lambda_\theta p'.s')u'$, and Δ preserves 11, 12 and 2. This implies $p \twoheadrightarrow p'$ and $u \twoheadrightarrow u'$, cfr. Lem. 3.5.20:(i), so that Lem. 3.4.12 implies $\{p'/_\theta\ u'\}$ decided, contradicting t' being a normal form. Thus we conclude.

  Assume $t = (\lambda_\theta p.s)u$, $\{p/_\theta\ u\} = \texttt{wait}$ and $\langle G, D\rangle = \mathcal{SM}_\theta(p, u) \neq \langle \varnothing, \varnothing\rangle$. We define Δ' as follows. If Δ includes the contraction of, at least, one head step, i.e. if there exists some $n \leqslant |\Delta|$ verifying $\langle \texttt{tgt}(\Delta[1..n-1]), \epsilon\rangle \in \Delta[n]$, we consider the minimum such n and define $\Delta' := \Delta[1..n-1]$. Otherwise, $\Delta' := \Delta$. In both cases $t \xrightarrow{\Delta'}\!\!\!\!\rightarrow (\lambda_\theta p'.s')u'$ and Δ' preserves 11 and 2, so that Lem. 3.5.20:(i) implies $p \xrightarrow{\Delta'|_{11}}\!\!\!\!\rightarrow p'$ and $u \xrightarrow{\Delta'|_2}\!\!\!\!\rightarrow u'$. Notice that in the latter case, $p', u' \in \boldsymbol{NF}$. In both cases we obtain that $\Delta'|_{11}$ uses G_r or $\Delta'|_2$ uses D_r, if $\{p'/_\theta\ u'\}$ decided by Lem. 3.5.22:(iii), otherwise by the IH:(iii). Recalling that in this case, $\mathcal{S}_\pi(t) = 11G \cup 2D$, we conclude by applying Lem. 3.5.20:(iii).

  Assume $t = (\lambda_\theta p.s)u$, $\{p/_\theta\ u\} = \texttt{wait}$, and $\mathcal{SM}_\theta(p, u) = \langle \varnothing, \varnothing\rangle$. A simple argument by contradiction based on Lem. 3.5.22:(iii) implies that $t' = (\lambda_\theta p'.s')u'$ and Δ preserves 11, 12 and 2. Therefore, Lem. 3.5.20:(i) implies $p \xrightarrow{\Delta|_{11}}\!\!\!\!\rightarrow p'$ and similarly for s and u. If $p \notin \boldsymbol{NF}$, so that $\mathcal{S}_\pi(t) = 11\mathcal{S}_\pi(p)$, then the IH:(i) can be applied to obtain that $\Delta|_{11}$ uses $\mathcal{S}(p)$, so that Lem. 3.5.20:(iii) allows to conclude. The remaining cases, i.e. $p \in \boldsymbol{NF}, s \notin \boldsymbol{NF}$ and $p, s \in \boldsymbol{NF}$ respectively, can be handled similarly.

  Assume $t = su$ and $s \notin \boldsymbol{ABS}$. If there exists some n such that $\texttt{tgt}(\Delta[n]) = s'u'$ and $s' \in \boldsymbol{ABS}$, then we consider the minimal such n, and let $\Delta' = \Delta[1..n]$. It is easy to deduce that Δ' preserves 1 and 2, so that Lem. 3.5.20:(i) implies $s \xrightarrow{\Delta'|_1}\!\!\!\!\rightarrow s'$. Observe that $s \notin \boldsymbol{NF}$, implying $\mathcal{S}_\pi(t) = 1\mathcal{S}_\pi(s)$. Moreover, $s \in \boldsymbol{DS}$ would imply $s' \in \boldsymbol{DS}$, so that $s \notin \boldsymbol{MF}$. Hence, a projection argument similar to that used in previous cases, based on Lem. 3.5.22:(i), allows us to conclude. Otherwise s does not reduce to an abstraction, implying $t = s'u'$, Δ preserves 1 and 2, and

$s', u' \in \boldsymbol{NF}$. Again, a projection argument applies, to $s \xrightarrow{\Delta|_1} s'$ if $s \notin \boldsymbol{NF}$, to $u \xrightarrow{\Delta|_2} u'$ otherwise, based on IH:(i).

Assume $t = \lambda_\theta p.s$. Then, $t' = (\lambda_\theta p'.s')$, Δ preserves 1 and 2, and $p', s' \in \boldsymbol{NF}$. A projection argument based on IH:(i) applies to $p \xrightarrow{\Delta|_1} p'$ or $s \xrightarrow{\Delta|_2} s'$, depending on whether $p \in \boldsymbol{NF}$ or not.

- Item (ii). Assume $p \notin \boldsymbol{MF}$, so that $G = \mathcal{S}_\pi(p)$ and $D = \varnothing$. The hypotheses imply $\mathcal{S}_\pi(p) \neq \varnothing$, and then p is not a normal form. Therefore, item (i) just proved applies to $p \xrightarrow{\Gamma} p'$, which suffices to conclude.

  Assume $p \in \boldsymbol{MF}$, $\neg bm(p, \theta)$, $u \notin \boldsymbol{MF}$. In this case, $G = \varnothing$ and $D = \mathcal{S}_\pi(u)$. Hence, an argument similar to that of the previous case applies on $u \xrightarrow{\Pi} u'$.

  Assume $p, u \in \boldsymbol{MF}$. In this case, $\{\!\{p \rhd_\theta u\}\!\} = \texttt{wait}$ implies $p = p_1 p_2$ and $u = u_1 u_2$, so that $G = 1G_1 \cup 2G_2$ and $D = 1D_1 \cup 2D_2$, where $\mathcal{SM}_\theta(p_i, u_i) = \langle G_i, D_i \rangle$ for $i = 1, 2$. The assumption $p, u \in \boldsymbol{MF}$ also implies $p' = p'_1 p'_2$, $u' = u'_1 u'_2$, and both Γ and Π preserve 1 and 2. Then Lem. 3.5.20:(i) implies $p_i \xrightarrow{\Gamma|_i} p'_i$ and $u_i \xrightarrow{\Pi|_i} u'_i$ for $i = 1, 2$. Moreover, $\langle G, D \rangle \neq \langle \varnothing, \varnothing \rangle$ implies $\langle G_k, D_k \rangle \neq \langle \varnothing, \varnothing \rangle$ for some $k \in \{1, 2\}$. Notice that $\{\!\{p_k \rhd_\theta u_k\}\!\}$ being positive (resp. $\texttt{fail}$) contradicts Lem. 3.5.21 (resp. $\{\!\{p \rhd_\theta u\}\!\} = \texttt{wait}$). Then $\{\!\{p_k \rhd_\theta u_k\}\!\} = \texttt{wait}$, so that either the IH (ii) or Lem. 3.5.22:(ii) applies, depending on whether $\{\!\{p'_k \rhd_\theta u'_k\}\!\}$ is $\texttt{wait}$ or positive. In either case, we obtain that $\Gamma|_k$ uses $(G_k)_r$, or $\Pi|_k$ uses $(D_k)_r$. Thus Lem. 3.5.20:(iii) allows to conclude.

$\square$

Lemma 3.5.24. *Let* $t \xrightarrow{\Delta} t'$, $b_r \in \mathcal{S}(t)[\![\Delta]\!]$, *and* a_r *verifying* $a_r < b_r$. *Then* a_r *is a matching failure.*

Proof. We prove the following, more general statement.

(i) The lemma statement.

(ii) Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$ such that $\{\!\{p \rhd_\theta u\}\!\} = \texttt{wait}$, $b_r \in G_r[\![\Gamma]\!]$ or $b_r \in D_r[\![\Pi]\!]$ where $\mathcal{SM}_\theta(p, u) = \langle G, D \rangle$, and a_r verifying $a_r < b_r$. Then a_r is a matching failure.

(iii) Let $p \xrightarrow{\Gamma} p'$ and $u \xrightarrow{\Pi} u'$ such that $\{p /_\theta u\} = \texttt{wait}$, $b_r \in G_r[\![\Gamma]\!]$ or $b_r \in D_r[\![\Pi]\!]$ where $\mathcal{SM}_\theta(p, u) = \langle G, D \rangle$, and a_r verifying $a_r < b_r$. Then a_r is a matching failure.

As in Lem. 3.5.22, item (iii) follows from item (ii). So we prove the others, by induction on $|t| + |u| + |p|$.

- We prove item (i). If t is either a variable or a matchable, then t is a normal form, contradicting the existence of b_r.

  Assume $t = (\lambda_\theta p.s)u$ and $\{p /_\theta u\}$ decided, implying $\mathcal{S}(t) = \{\langle t, \epsilon \rangle\}$. Then, a straightforward inductive argument on $|\Delta|$ yields that $\mathcal{S}(t)[\![\Delta]\!] = \varnothing$ or $b_r = \langle t', \epsilon \rangle$, contradicting in both cases the existence of a_r. Thus we conclude.

  Assume $t = (\lambda_\theta p.s)u$, $\{p /_\theta u\} = \texttt{wait}$, and $\langle G, D \rangle = \mathcal{SM}_\theta(p, u) \neq \langle \varnothing, \varnothing \rangle$. Then $\mathcal{S}_\pi(t) = 11G \cup 2D$. Consider Δ', Δ'' such that $\Delta = \Delta'; \Delta''$, $t \xrightarrow{\Delta'} t'' = (\lambda_\theta p'.s')u' \xrightarrow{\Delta''}$

t', Δ' preserves 11 and 2, and either $\Delta'' = \mathtt{nil}_{t''}$ or $\langle t'', \epsilon \rangle \in \Delta''[1]$. Lem. 3.5.20:(i) implies $p \xrightarrow{\Delta'|_{11}} p'$ and $u \xrightarrow{\Delta'|_2} u'$. If $\{p'/_\theta\ u'\}$ is positive, then Lem. 3.5.22:(iii) implies $G_r[\![\Delta'|_{11}]\!] = D_r[\![\Delta'|_2]\!] = \varnothing$, and therefore Lem. 3.5.20:(ii) yields $\mathcal{S}(t)[\![\Delta']\!] = \varnothing$. If $\{p'/_\theta\ u'\} = \mathtt{fail}$ and $\Delta'' \neq \mathtt{nil}_{t''}$, then it is immediate to obtain $t' = I$, a normal form, contradicting the existence of b_r. Therefore, assume $\{p'/_\theta\ u'\} \in \{\mathtt{wait}, \mathtt{fail}\}$ and $\Delta'' = \mathtt{nil}_{t''}$, so that $\Delta = \Delta'$ and $t' = (\lambda_\theta p'.s')u'$. An analysis of the ancestor of b_r, which is some $b_{0r} \in \mathcal{S}(t)$, along with Lem. 3.5.20:(ii), yields that $b = 11b'$ where $b'_r \in G_r[\![\Delta\,|_{11}]\!]$ or $b = 2b'$ where $b'_r \in D_r[\![\Delta\,|_2]\!]$, implying respectively that $b'_r \in \mathcal{RO}(p')$ or $b'_r \in \mathcal{RO}(u')$. Let a_r verifying $a_r < b_r$. If $a = \epsilon$, then $\{p'/_\theta\ u'\} = \mathtt{fail}$, i.e. a_r is a matching failure. Otherwise, $a = 11a'$ or $a = 2a'$, so that $a'_r \in \mathcal{RO}(p')$ or $a'_r \in \mathcal{RO}(u')$ respectively, and $a'_r < b'_r$. Therefore IH (iii) implies that a'_r is a matching failure, which suffices to conclude.

Assume $t = (\lambda_\theta p.s)u$, $\{p/_\theta\ u\} = \mathtt{wait}$, and $\mathcal{SM}_\theta(p,u) = \langle \varnothing, \varnothing \rangle$. Observe that $t \xrightarrow{\Gamma} (\lambda_\theta p''.s'')u''$ such that $\{p''/_\theta\ u''\}$ is decided would contradict $\mathcal{SM}_\theta(p,u) = \langle \varnothing, \varnothing \rangle$; cfr. Lem. 3.5.20:(i) and Lem. 3.5.22:(iii) considering a minimal such Γ. Therefore, $t' = (\lambda_\theta p'.s')u'$, Δ preserves 11, 12 and 2, and $\{p'/_\theta\ u'\} = \mathtt{wait}$. If $p \notin \boldsymbol{NF}$, so that $\mathcal{S}_\pi(t) = 11\mathcal{S}_\pi(p)$, then Lem. 3.5.20:(i) and Lem. 3.5.20:(ii) imply $p \xrightarrow{\Delta|_{11}} p'$ and $b = 11b'$ where $b'_r \in \mathcal{S}(p)[\![\Delta|_{11}]\!]$ respectively. Observe $\{p'/_\theta\ u'\} = \mathtt{wait}$ implies that $\langle t', \epsilon \rangle \notin \mathcal{RO}(t')$. Then $a_r < b_r$ implies $a = 11a'$, so that $a'_r \in \mathcal{RO}(p')$, and $a'_r < b'_r$. Hence the IH:(i) applies, which suffices to conclude. The other cases ($p' \in \boldsymbol{NF}$ and $s' \notin \boldsymbol{NF}$, and $p', s' \in \boldsymbol{NF}$) admit analogous arguments.

Assume $t = su$ where $s \notin \boldsymbol{ABS}$. Let Δ', Δ'' such that $\Delta = \Delta'; \Delta''$, $t \xrightarrow{\Delta'} s'u' \xrightarrow{\Delta''} t'$, Δ' preserves 1 and 2, and either $\Delta'' = \mathtt{nil}_{t'}$ or $\langle s'u', \epsilon \rangle \in \Delta''[1]$. Lem. 3.5.20:(i) implies $s \xrightarrow{\Delta'|_1} s'$ and $u \xrightarrow{\Delta'|_2} u'$.

- If $s' \in \boldsymbol{ABS}$, then $s \neq s'$ implying that s is not a normal form, and therefore $\mathcal{S}_\pi(t) = 1\mathcal{S}_\pi(s)$. Moreover, $s \notin \boldsymbol{MF}$; notice that $s \in \boldsymbol{DS}$ would imply $s' \in \boldsymbol{DS}$. Therefore, Lem. 3.5.22:(iii) implies $\mathcal{S}(s)[\![\Delta'|_1]\!] = \varnothing$, so that Lem. 3.5.20:(ii) contradicts the existence of b_r. Thus we conclude.
- If $s' \notin \boldsymbol{ABS}$, then $\Delta'' = \mathtt{nil}_{s'u'}$, so that $\Delta = \Delta'$ and $t' = s'u'$. Moreover, $\langle t', \epsilon \rangle \notin \mathcal{RO}(t')$. If s is not a normal form, so that $\mathcal{S}_\pi(t) = 1\mathcal{S}_\pi(s)$, then Lem. 3.5.20:(ii) implies $b = 1b'$ where $b'_r \in \mathcal{S}(s)[\![\Delta\,|_1]\!]$. On the other hand, $a_r < b_r$ implies $a = 1a'$ where $a'_r \in \mathcal{RO}(s')$. Then the IH:(i) applies, which suffices to conclude. If s is a normal form, so that $\mathcal{S}_\pi(t) = 2\mathcal{S}_\pi(u)$, then a similar argument applies.

Assume $t = \lambda_\theta p.s$. Then Δ preserves 1 and 2, so that $t' = \lambda_\theta p'.s'$ and Lem. 3.5.20:(i) implies $p \xrightarrow{\Delta|_1} p'$ and $s \xrightarrow{\Delta|_2} s'$. A projection argument based on IH (i) analogous to those used in previous cases, on $p \xrightarrow{\Delta|_1} p'$ or $s \xrightarrow{\Delta|_2} s'$ depending whether $p \in \boldsymbol{NF}$, allows to conclude.

- We prove item (ii). There are three cases to analyse, given $\{\!\{p \rhd_\theta u\}\!\} = \mathtt{wait}$.

  If $p \notin \boldsymbol{MF}$, then $G_r = \mathcal{S}(p)$ and $D_r = \varnothing$, so that $b_r \in G_r[\![\Gamma]\!] = \mathcal{S}(p)[\![\Gamma]\!]$. The IH:(i) on $p \xrightarrow{\Gamma} p'$ suffices to conclude.

If $p \in \boldsymbol{MF}$ and $u \notin \boldsymbol{MF}$, so that $G_r = \varnothing$ and $D_r = \mathcal{S}(u)$, then an analogous argument applies.

If $p = p_1 p_2$, $u = u_1 u_2$, and $p, u \in \boldsymbol{MF}$, then $G = 1G_1 \cup 2G_2$ and $D = 1D_1 \cup 2D_2$, where $\langle G_i, D_i \rangle = \mathcal{SM}_\theta(p_i, u_i)$ for $i = 1, 2$. Moreover, $p, u \in \boldsymbol{MF}$ implies that Γ and Π preserve 1 and 2, $p' = p'_1 p'_2$ and $u' = u'_1 u'_2$. Lem. 3.5.20:(i) yields $p_i \xrightarrow{\Gamma|_i} p'_i$ and $u_i \xrightarrow{\Pi|_i} u'_i$ for $i = 1, 2$. In turn, Lem. 3.5.20:(ii) implies $b = kb'$ where $b'_r \in G_{kr}[\![\Gamma|_k]\!]$ or $b'_r \in D_{kr}[\![\Pi|_k]\!]$, for some $k \in \{1, 2\}$. We consider that k. Observe that $\{\!\{p_k \rhd_\theta u_k\}\!\} = \texttt{fail}$ would contradict $\{\!\{p \rhd_\theta u\}\!\} = \texttt{wait}$, and $\{\!\{p_k \rhd_\theta u_k\}\!\}$ positive would imply $G_k = D_k = \varnothing$ by Lem. 3.5.21. Therefore $\{\!\{p_k \rhd_\theta u_k\}\!\} = \texttt{wait}$. Observe that neither $\langle p', \epsilon \rangle$ nor $\langle u', \epsilon \rangle$ are steps, so that $a_r < b_r$ and $b = kb'$ imply $a = ka'$. Hence the IH:(ii), applied on $p_k \xrightarrow{\Gamma|_k} p'_k$ and $u_k \xrightarrow{\Pi|_k} u'_k$, allows to conclude. □

Proposition 3.5.25. *Let t be a term not in normal form. Then $\mathcal{S}(t)$ is a non-gripping set.*

Proof. Let $t \xrightarrow{\Psi} u$, $a_r \in \mathcal{RO}(u)$, $b_r \in \mathcal{S}(t)[\![\Psi]\!]$; it suffices to deduce that b_r does not grip a_r. If $a_r \nless b_r$, then we immediately conclude. If $a_r < b_r$, then Lem. 3.5.24 entails that a_r is a matching failure so b_r cannot grip a_r. □

Theorem 3.5.26. *The reduction strategy $\mathcal{S}$, cfr.Dfn. 3.5.2, is normalising.*

Proof. The results in Section 3.4.3 yield that PPC enjoys all the required axioms. Prop. 3.5.23 and Prop. 3.5.25 imply that the sets of steps selected by $\mathcal{S}$ are necessary and non-gripping. Thus the statement is an immediate consequence of Thm. 3.3.14. □

Chapter 4

Standardisation for the linear substitution calculus

As we described in the introduction, cfr. Section 1.2.2, the **linear substitution calculus**, notation $\lambda_{\mathtt{lsub}}^{\sim}$, is an explicit substitution (ES) calculus *at a distance*, in which substitutions are not propagated along terms, thus yielding a simpler reduction space than that of previously proposed ES calculi.

The linear substitution calculus enjoys several properties expected for ES calculi. The proofs can be obtained as minor variations of the proofs given for other ES calculi at a distance, cfr. [KÓC08, AK10].

The purpose of this chapter is to establish standardisation results for $\lambda_{\mathtt{lsub}}^{\sim}$, through the ARS model. To reach this goal, it suffices to model this calculus as an ARS, and show that the resulting ARS satisfies all the axioms required by the abstract standardisation theorems presented in [Mel96], which are included in this thesis as Thm. 2.1.23 and Thm. 2.1.24, cfr.page 41. There are three aspects which make the characterisation of $\lambda_{\mathtt{lsub}}^{\sim}$ as an ARS a non-trivial task.

Firstly, the *identification of steps*, and thus the definition of *residual relation*, is not immediate, because different coinitial steps could correspond to the same subterm. Recall the rule

$$C[\![x]\!]\,[x/u] \;\longrightarrow\; C[\![u]\!]\,[x/u] \tag{4.1}$$

introduced in page 11, and consider the two steps for this rule in the term $(xx)\,[x/y]$: these steps can be distinguished only by the respective occurrence of x.

Secondly, the intent of the *embedding relation* in the ARS model, as described in Section 1.3.1, is not coherent with the *syntactic nesting* of steps. We recall that, according to the ARS model, $a < b$ should indicate that the contraction of a could possibly result in the erasure, or the duplication, of b. Cfr. the axiom Linearity, defined in Section 2.1.5. As discussed in Section 2.1.1, for the λ-calculus, this criterion is coherent with the nesting of steps: a necessary condition for a step a to erase or duplicate a step b, is that (the subterm corresponding to) a nests (that) b in their common source term.

This coherence does not hold in $\lambda_{\mathtt{lsub}}^{\sim}$. Recall the rule (4.1), and let us remark that the explicit substitution construct $s\,[x/u]$ binds the occurrences of x in s. Then term

$$t = (\underline{x}z)\,[x/\underline{y}]\,[y/u]$$

includes two steps corresponding to the underlined occurrences of x and y. Let us call these steps a_x and a_y respectively. Let us find out the subterms corresponding to a_x and a_y. The scope of the substitution $[x/\underline{y}]$ is the subterm $(\underline{x}z)$; therefore, the context C indicated in (4.1) for a_x is $(\Box z)$. In turn, the scope of the substitution $[y/u]$ is the subterm $(\underline{x}z)[x/\underline{y}]$, notice that t is properly parsed as the added parentheses in $((\underline{x}z)[x/\underline{y}])[y/u]$ suggest. Hence, the context C for a_y is $(\underline{x}z)[x/\Box]$. Consequently, we can mark the subterms corresponding to the steps a_x and a_y as follows:

$$t = \overbrace{(\underbrace{(\underline{x}z)[x/\underline{y}]}_{a_x})\,[y/u]}^{a_y}$$

The contraction of these steps yields

$$(\underline{x}z)[x/\underline{y}][y/u] \xrightarrow{a_x} (\underline{y}z)[x/\underline{y}][y/u] = t_x \qquad (\underline{x}z)[x/\underline{y}][y/u] \xrightarrow{a_y} (\underline{x}z)[x/u][y/u] = t_y$$

Notice that a_y has *two* residuals after a_x, matching the two underlined occurrences of y in t_x.[1] On the other hand a_x has exactly one residual after a_y. Therefore, any definition of the embedding relation in an ARS modeling $\lambda_{\widetilde{\mathtt{lsub}}}$ should establish $a_x < a_y$, whereas the nesting between the corresponding subterms goes in the opposite direction.

The third aspect which makes modeling of $\lambda_{\widetilde{\mathtt{lsub}}}$ as an ARS complex, is the *correspondence between the calculus and the linear logic proof-nets* [Gir87].

Terms of $\lambda_{\widetilde{\mathtt{lsub}}}$ and proof-nets are behaviorally equivalent in a strong sense: every term t maps to a proof-net P_t, and every *step* on t or P_t maps to an evaluation *step* on the other. Additionally, a *bijection* can be established between the steps in t and those in P_t, so that concepts such as residuals transfer from terms to proof-nets and vice versa. As expressed in Section 1.2.2, an equivalence relation, defined through equational logic from three simple equations, turns the behavioral equivalence between terms and proof-nets into a true *isomorphism*: each proof-net corresponds exactly to a class of equivalent terms. From this perspective, we could consider equivalent terms as just different representations of an unique object being rewritten.[2] On the other hand, to support this view, we should be able to define an ARS whose objects were not single terms, but classes of equivalent terms. Therefore, all the elements participating in an ARS, as steps, residuals and embedding, should be stable by this equivalence.

We define a first ARS for $\lambda_{\widetilde{\mathtt{lsub}}}$, using *labels* to identify and trace steps, leading to the definition of the residual relation, and using a simple *left-to-right* embedding order on coinitial steps. This ARS enjoy all the axioms required in Thm. 2.1.24, namely the initial axioms, FD, SO, and all the embedding axioms. We remark that this calculus enjoy *semantical orthogonality*, despite the fact that it fails to comply with a syntactic orthogonality criterion: e.g., the two steps in $(xx)[x/y]$ form a critical pair.[3] We also notice that the left-to-right embedding imposes a total order on coinitial steps, making it easier to verify the embedding axioms. We obtain a strong standardisation result for this ARS: each class of equivalent reduction sequences includes exactly one `s.r.s.`.

[1] As in a previous example, these two steps correspond to the same subterm of t_x, being distinguished only by the contexts of the respective occurrences of y.

[2] as it is the case w.r.t. α-equivalence in the λ-calculus, cfr. [LV02].

[3] We discuss different perspectives about orthogonality in Section 6.2.3.

W.r.t. the aspects making the characterisation of $\lambda_{\mathtt{lsub}}^{\sim}$ in the ARS model a challenging task, the two former ones are adequately addressed: labels lead to a sound definition of residuals, and the left-to-right order is coherent with the intent of the embedding relation. On the other hand, this is not the case for the last concern, i.e., stability by equivalence on terms. Particularly, the left-to-right-order on coinitial labels is not stable. Recall the equation described in Section 1.2.2:

$$t[x/s][y/u] \;\approx\; t[y/u][x/s] \qquad\qquad \text{if } x \notin \mathtt{fv}(u) \text{ and } y \notin \mathtt{fv}(s)$$

and consider the following terms:

$$t = (yx)[x/s_1][y/s_2] \sim (yx)[y/s_2][x/s_1] = t'$$

It is clear that the relative embedding between two steps, lying inside s_1 and s_2 respectively, are different in t than in t'.

To deal with the equivalence defined on terms, we show that labels induce, for any pair of equivalent terms, a bijection between their sets of steps such that targets of related steps are again equivalent. Moreover, the label-based characterisation of residuals preserves these bijections: residuals of related steps in equivalent terms, after related steps, are again related. Furthermore, we define an order on coinitial steps, the *box order*, which is stable by equivalence on terms, via the bijection on steps just mentioned.

These elements allow to define a second ARS for $\lambda_{\mathtt{lsub}}^{\sim}$, which satisfies the three concerns identified for the characterisation of the calculus in the ARS model. This ARS satisfies all the axioms required in Thm. 2.1.23, yielding the existence of a `s.r.s.` equivalent to any reduction sequence. Unfortunately, this ARS fails two axioms required for Thm. 2.1.24, so that uniqueness of `s.r.s.` in each class of equivalent reduction sequences cannot be derived from this result. We nevertheless obtain uniqueness of `s.r.s.` for this ARS, and hence for $\lambda_{\mathtt{lsub}}^{\sim}$, by developing a *novel abstract standardisation proof* in the ARS model. This proof makes use of the existence of two different ARS modeling the same rewriting system.

Therefore, the material in this chapter makes a contribution to the study of explicit substitution calculi at a distance, and also a contribution to the ARS model. These results were presented in [ABKL14], along with other results about $\lambda_{\mathtt{lsub}}^{\sim}$. We mention a coinductive characterisation of the notion of *external* step, which allows to show that the leftmost reduction strategy is normalising for this calculus. The *linear head reduction* for $\lambda_{\mathtt{lsub}}^{\sim}$ is also studied in that work.

Plan of the chapter

In Section 4.1 we introduce $\lambda_{\mathtt{lsub}}^{\sim}$, following [ABKL14]. The labels used to identify and trace steps, and the *graphical equivalence* on terms which sets the isomorphism with proof-nets, are presented as well. In Section 4.2 we describe the first model of $\lambda_{\mathtt{lsub}}^{\sim}$ as an ARS, showing that it verifies all the axioms required in the standardisation theorems stated in Section 2.1.8. Therefore, our first standardisation results are given in Section 4.3, where we discuss also its limitations. Section 4.4 is devoted to show that the label-based definition of steps and residuals is stable by the graphical equivalence. In Section 4.5 we define the *box order* and use it to define the second ARS which models $\lambda_{\mathtt{lsub}}^{\sim}$; we verify the axioms needed to obtain the existence of an `s.r.s.` equivalent to any given reduction sequence, and subsequently show why the stronger uniqueness result cannot be obtained through the abstract standardisation results presented in [Mel96]. Finally, in Section 4.6 we develop a novel abstract standardisation proof, and apply it to $\lambda_{\mathtt{lsub}}^{\sim}$.

4.1 The linear substitution calculus

Consider a countable set of *variables* $x, y, z, x', x_1, \ldots$. The set of terms of the *linear substitution calculus*, denoted by $\mathcal{T}$, is generated by the following grammar:

$$t \ ::= \ x \ | \ tt \ | \ \lambda x.t \ | \ t[x/t]$$

A term x is called a **variable**, tu an **application**, $\lambda x.t$ an **abstraction** and $t[x/u]$ an **explicit substitution**. We will use $\mathtt{L}, \mathtt{L}', \mathtt{L}_1, \ldots$ to denote (possibly empty) **lists of substitutions** $[x_1/t_1] \ldots [x_n/t_n]$. Notice that the lists of substitutions are not terms.

In the terms $\lambda x.t$ and $t[x/u]$, the occurrences of x in t are **bound** by the abstraction or the explicit substitution respectively. The derived notion of free variables of a term t is denoted by $\mathtt{fv}(t)$. As usual, we consider terms up to α-conversion, i.e. up to renaming of bound variables. When needed, we will assume that terms follow Barendregt's *variable convention*, cfr. [Bar84].

A **context** is a term having at least one occurrence of a designated symbol $\Box$ called the *hole*. In this chapter we use mostly contexts having exactly one occurrence of the hole, i.e. *one-hole contexts*. We also use occasionally *two-* and *three-hole contexts*. The meta-notation $\mathtt{L} = [x_1/t_1] \ldots [x_k/t_k]$ can also be seen as a context $\Box[x_1/t_1] \ldots [x_k/t_k]$. We write $C[t]$ for the term obtained by replacing the only hole of a one-hole context C by the term t. We write $C[\![u]\!]$ when the free variables of u are not captured by the context C, i.e. there are no abstractions or explicit substitutions in C that bind the variables of $\mathtt{fv}(u)$. We use $D[t_1, t_2]$ (resp. $D[t_1, t_2, t_3]$) analogously to $C[t]$ for a two-hole (resp. three-hole) context D.

In the following, we give the definition of two variants of the linear substitution calculus, which we call $\lambda_{\mathtt{lsub}}$ and $\lambda_{\mathtt{lsub}}^{\sim}$ respectively.

Definition 4.1.1 ($\lambda_{\mathtt{lsub}}$-calculus)**.** *The* $\lambda_{\mathtt{lsub}}$***-calculus*** *is defined by the preceding syntax, plus the* semantics *given by the* ***reduction relation*** $\rightarrow_{\lambda_{\mathtt{lsub}}}$. *This relation is defined as the union of* $\rightarrow_{\mathtt{db}}$, $\rightarrow_{\mathtt{ls}}$, *and* $\rightarrow_{\mathtt{gc}}$, *which are the closure by contexts* C *of the following rewriting rules:*

$$\begin{array}{lll} (\lambda x.t)\mathtt{L}u & \mapsto_{\mathtt{db}} & t[x/u]\mathtt{L} \\ C[\![x]\!][x/u] & \mapsto_{\mathtt{ls}} & C[\![u]\!][x/u] \\ t[x/u] & \mapsto_{\mathtt{gc}} & t \qquad\qquad \textit{if } x \notin \mathtt{fv}(t) \end{array}$$

The names db, ls, *and* gc *stand for* distant beta, linear substitution, *and* garbage collection, *respectively.*

Rule $\mapsto_{\mathtt{db}}$ (resp. $\mapsto_{\mathtt{ls}}$) comes from the structural λ-calculus [AK10] (resp. Milner's calculus [Mil07b]), while $\mapsto_{\mathtt{gc}}$ belongs to both calculi. In db we may assume w.l.o.g. that $\bigcup_{i=1}^{k}\{x_i\} \cap \mathtt{fv}(u) = \varnothing$ and $x \notin \mathtt{fv}(u) \cup \bigcup_{i=1}^{k} \mathtt{fv}(t_i)$. The occurrence of L, considered as a context, in the db-rule, the use of a context C in the ls-rule, and the global side condition in the gc-rule, justify the idea of rewriting rules *at a distance.*

Fig. 4.1 shows a reduction sequence including applications of the three rules. We remark the application at a distance of the db- and ls-rules in the second and third steps, respectively. In turn, the application of the $\mathtt{gc}$-rule in the last step also act at a distance: there is no need to propagate the substitution $[x/\mathtt{a}]$ through the term prior to garbage-collecting it. Cfr. the $\lambda\mathtt{x}$ and the $\lambda_{\mathtt{lsub}}^{\sim}$ reduction sequence examples given in Section 1.2.2, Figs. 1.2.2 and 1.4 respectively.

$$\begin{array}{lll} (\lambda x.(\lambda y.y(xy)))\mathtt{ab} & \rightarrow_{\mathtt{db}} & (\lambda y.y(xy))[x/\mathtt{a}]\mathtt{b} \\ & \rightarrow_{\mathtt{db}} & (y(xy))[y/\mathtt{b}][x/\mathtt{a}] \\ & \rightarrow_{\mathtt{ls}} & (y(\mathtt{a}y))[y/\mathtt{b}][x/\mathtt{a}] \\ & \rightarrow_{\mathtt{gc}} & (y(\mathtt{a}y))[y/\mathtt{b}] \end{array}$$

Figure 4.1: An example reduction sequence in $\lambda_{\mathtt{lsub}}$

The list-of-substitutions context $\mathtt{L}$ in rule $\mathtt{db}$ is motivated by its encoding in proof-nets, where substitutions are in fact unordered, except for occurrences of variables bound by a substitution in the list. Moreover, in proof-nets substitutions are partially free to float (i.e. to traverse some term constructors). These features of substitutions are formalized as follows:

Definition 4.1.2 (Graphical equivalence)**.** *We define the* ***graphical equivalence****, notation $\sim$, as the contextual, transitive, symmetric and reflexive closure of α-conversion and the following axioms:*

$$\begin{array}{llll} t[x/u][y/s] & \approx_{\mathtt{CS}} & t[y/s][x/u] & x \notin \mathtt{fv}(s)\ \&\ y \notin \mathtt{fv}(u) \\ (\lambda y.t)[x/u] & \approx_{\sigma_1} & \lambda y.t[x/u] & y \notin \mathtt{fv}(u) \\ (ts)[x/u] & \approx_{\sigma_2} & t[x/u]s & x \notin \mathtt{fv}(s) \end{array}$$

This equivalence characterizes exactly the representation of terms as proof-nets, in the sense that $t \sim u$ iff t and u map to the same proof-net [Acc11]. We use $\approx$ to denote the union of $\approx_{\mathtt{CS}}$, $\approx_{\sigma_1}$ and $\approx_{\sigma_2}$, and $\overset{1}{\sim}$ to denote the symmetrical and contextual closure of the union of α-conversion and $\approx$. Therefore, $\sim$ is the reflexive and transitive closure of $\overset{1}{\sim}$.

Definition 4.1.3 ($\lambda_{\mathtt{lsub}}^{\sim}$-calculus)**.** *The* $\lambda_{\mathtt{lsub}}^{\sim}$***-calculus*** *is given by the set of terms $\mathcal{T}$ modulo the graphical equivalence $\sim$, and by the* ***reduction relation*** $\rightarrow_{\lambda_{\mathtt{lsub}}^{\sim}}$*, defined as follows: $\lfloor t \rfloor \rightarrow_{\lambda_{\mathtt{lsub}}^{\sim}} \lfloor u \rfloor$ iff there exist t', u' verifying $t \sim t' \rightarrow_{\lambda_{\mathtt{lsub}}} u' \sim u$, where $\lfloor t \rfloor$ denotes the $\sim$-equivalence class of terms associated to t. Thus in particular $t \rightarrow_{\lambda_{\mathtt{lsub}}} t'$ implies $\lfloor t \rfloor \rightarrow_{\lambda_{\mathtt{lsub}}^{\sim}} \lfloor t' \rfloor$.*

Notice that the terms related by the graphical equivalence equations σ_1 and σ_2 are syntactically different: the main construct in $(\lambda y.t)[x/u]$ is the *explicit substitution*, while it is the *abstraction* for $\lambda y.t[x/u]$; a similar situation can be seen in σ_2. In turn, the terms related by the $\mathtt{CS}$ equation are also syntactically different in a remarkable way: they are parsed as $(t[x/u])[y/s]$ and $(t[y/s])[x/u]$ respectively. We notice that the list-of-substitutions meta-notation $\mathtt{L}$ is at odds with syntactic structure: if $\mathtt{L} = [x_1/t_1]\ldots[x_n/t_n]$ and u is a term, then the term that we write informally as $u\mathtt{L}$ actually parses as $((u[x_1/t_1])\ldots)[x_n/t_n]$. The intuition given by the graphical equivalence and the $\mathtt{L}$ meta-notation, corresponding with proof-nets, is in tension with the term syntax. This tension motivates the careful study of the properties of $\lambda_{\mathtt{lsub}}^{\sim}$ that we perform in Section 4.4. This study entails that $\lambda_{\mathtt{lsub}}^{\sim}$ behaves as expected.

4.1.1 A labeled version

In Sections 4.2 and 4.5.1, we introduce several ARSs to model $\lambda_{\mathtt{lsub}}$ and $\lambda_{\widetilde{\mathtt{lsub}}}$. Residuals are defined in these ARSs by *labeling* redexes. In order to compute $a[\![b]\!]$, the redex a is given a unique label, say α, obtaining a *labeled term*. This labeled term includes a step corresponding to b; the residuals of a after b are exactly the redexes labeled with α in the target of that step.

We formalise this idea by defining a labeled version of $\lambda_{\widetilde{\mathtt{lsub}}}$. Consider a countable set of **labels**, i.e. special symbols denoted as $\alpha, \beta, \gamma, \ldots$. The set of labeled terms, denoted by $\mathcal{T}_{\mathcal{L}}$, is generated by the following grammar.

$$t ::= x \mid x^{\alpha} \mid tt \mid \lambda x.t \mid \lambda x^{\alpha}.t \mid t[x/t] \mid t[x^{\alpha}/t]$$

The notations $x^{(\alpha)}$, $\lambda x^{(\alpha)}.t$ and $t[x^{(\alpha)}/t]$ mean that x may or may not be labeled. We write $\mathtt{Lab}(t)$ to denote the **set of all the labels** of t and t° to denote the term obtained from t by **removing all its labels**. Thus for example $((x^{\alpha}y^{\beta})[y/\lambda z^{\gamma}.z])^{\circ} = (xy)[y/\lambda z.z]$. A term t can be labeled in different ways, leading to different variants of t. More precisely, we say that t is a **variant** of u iff $t^{\circ} = u^{\circ}$. Thus in particular, t is a variant of itself.

We extend the meta-notation $\mathtt{L}$ to lists of possibly labeled substitutions, and C to possibly labeled contexts. Similarly, the notions of free and bound variables are extended to labeled terms as expected together with their corresponding notion of α-conversion. We use $\mathtt{flv}(t)$ to denote the subset of $\mathtt{fv}(t)$ having at least one labeled occurrence, e.g. $\mathtt{flv}((x^{\alpha}y^{\beta})[y/z]) = \{x\}$.

Definition 4.1.4. ***Labeled reduction*** $\stackrel{\alpha}{\to}$ *on labeled terms is defined as the contextual closure of the following rewriting rules, on labeled terms:*

$$\begin{array}{llll} (\lambda x^{\alpha}.t)\mathtt{L}u & \stackrel{\alpha}{\mapsto}_{\mathtt{db}} & t[x/u]\mathtt{L} & \\ C[\![x^{\alpha}]\!][x/u] & \stackrel{\alpha}{\mapsto}_{\mathtt{ls}} & C[\![u]\!][x/u] & \\ t[x^{\alpha}/u] & \stackrel{\alpha}{\mapsto}_{\mathtt{gc}} & t & x \notin \mathtt{fv}(t) \end{array}$$

Definition 4.1.5. *The* ***labeled graphical equivalence***[4] $\sim$ *on* labeled terms *is given by the contextual, transitive, symmetric and reflexive closure of α-conversion and the following axioms:*

$$\begin{array}{llll} t[x^{(\alpha)}/u][y^{(\beta)}/s] & \approx_{\mathtt{CS}} & t[y^{(\beta)}/s][x^{(\alpha)}/u] & x \notin \mathtt{fv}(s)\ \&\ y \notin \mathtt{fv}(u) \\ (\lambda y^{(\beta)}.t)[x^{(\alpha)}/u] & \approx_{\sigma_1} & \lambda y^{(\beta)}.t[x^{(\alpha)}/u] & y \notin \mathtt{fv}(u) \\ (ts)[x^{(\alpha)}/u] & \approx_{\sigma_2} & t[x^{(\alpha)}/u]s & x \notin \mathtt{fv}(s) \end{array}$$

The axioms are to be understood in such a way that each label occurs either in both sides of the axiom or in none of them.

Notice that the terms, lists of substitutions and contexts which occur in Dfns. 4.1.4 and 4.1.5 are labeled. In order to show that $\lambda_{\widetilde{\mathtt{lsub}}}$ satisfies the SO axiom, we will work with the following subset of labeled terms.

Definition 4.1.6. *We define the set of* ***well-labeled terms****, notation $\mathcal{T}_{\mathcal{WL}}$, as follows:*

[4]By abuse of notation we use the same symbol both for the equivalence relation on labeled and unlabeled terms.

- *$x \in \mathcal{T}_{\mathcal{WL}}$ and $x^\alpha \in \mathcal{T}_{\mathcal{WL}}$*
- *If $t \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \texttt{flv}(t)$ then $\lambda x.t \in \mathcal{T}_{\mathcal{WL}}$*
- *If $t, u \in \mathcal{T}_{\mathcal{WL}}$, then $tu \in \mathcal{T}_{\mathcal{WL}}$*
- *If $(\lambda x.t)\texttt{L}, u \in \mathcal{T}_{\mathcal{WL}}$, then $(\lambda x^\alpha.t)\texttt{L}u \in \mathcal{T}_{\mathcal{WL}}$*
- *If $t, u \in \mathcal{T}_{\mathcal{WL}}$, then $t[x/u] \in \mathcal{T}_{\mathcal{WL}}$*
- *If $t, u \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \texttt{fv}(t)$, then $t[x^\alpha/u] \in \mathcal{T}_{\mathcal{WL}}$.*

Note that $\lambda x^\alpha.x$, $\lambda x.x^\alpha$ and $x[x^\alpha/u]$ are not in $\mathcal{T}_{\mathcal{WL}}$. Note also that subterms of well-labeled terms are not necessarily well-labeled (e.g. the abstraction of a labeled db-redex). Well-labeled terms are stable by reduction and graphical equivalence:

Lemma 4.1.7. *Let $t \in \mathcal{T}_{\mathcal{WL}}$. If $t \xrightarrow{\alpha} u$ or $t \sim u$, then $u \in \mathcal{T}_{\mathcal{WL}}$.*

Proof. See Appendix C, page 263. □

4.2 A first ARS to model $\lambda_{\texttt{lsub}}$

In this section, we define an ARS $\mathfrak{A}_{\texttt{L}}$ to model the $\lambda_{\texttt{lsub}}$ calculus. This ARS is a first, approximate model of the reduction spaces of $\lambda_{\texttt{lsub}}$. It allows us to obtain some preliminary standardisation results. These results, in turn, are used in the study of more "accurate" models of the linear substitution calculus, given by the ARSs we define in Section 4.5.1. For the relation between the ARS $\mathfrak{A}_{\texttt{L}}$ and those we define later, cfr. Section 2.1.9.

Definition 4.2.1. *We define the ARS $\mathfrak{A}_{\texttt{L}} = \langle \mathcal{O}, \mathcal{R}, \mathsf{src}, \mathsf{tgt}, [\![\cdot]\!], \prec_{\texttt{L}} \rangle$ as follows.*

Objects

The set $\mathcal{O}$ of objects *is the set of terms of $\lambda_{\texttt{lsub}}$.*

Steps, source, target

There are three kinds of steps, corresponding to the three rules of $\lambda_{\texttt{lsub}}$. Let D be a context and r a term.

- *A pair $\langle D, r \rangle$ is a* `db`***-step*** *iff $r = (\lambda x.s)\texttt{L}u$,*
- *a triple $\langle D, r, C \rangle$ is a* `ls`***-step*** *iff $r = C[\![x]\!][x/u]$, and*
- *a pair $\langle D, r \rangle$ is a* `gc`***-step*** *iff $r = s[x/u]$ and $x \notin \texttt{fv}(s)$.*

The set $\mathcal{R}$ of ***steps*** *is the union of the sets of* `db`-, `ls`- *and* `gc`*-steps.*

For all kinds of steps, the ***source*** *is the term $D[r]$. The* ***target*** *is $D[s[x/u]\texttt{L}]$, $D[C[\![u]\!][x/u]]$, or $D[s]$, for* `db`-, `ls`- *and* `gc`*-steps respectively.*

The ***anchor*** *of a step is the variable occurrence which would possibly carry a label in a labeled variant of the corresponding term. It is the only occurrence of x for* `db`- *and* `gc`*-steps, and the one inside the context C for* `ls`*-steps.*

Residual relation

Observe that the definition of steps applies also to labeled *terms, exactly as it is given for unlabeled terms, provided that D, r and C stand for labeled contexts and terms. Let t be a labeled term, $a \in \mathcal{RO}(t)$ (where the anchor of a can be labeled or not), and $\alpha \notin \mathtt{Lab}(t)$. If t is a labeled term, a is a step and α is a label, then we say that a is an **α-labeled step** in t iff $\mathsf{src}(a) = t$ and the anchor of a is labeled with α. We define the **a-α-lift** of t, notation $\mathtt{lift}(t, a, \alpha)$, as the variant of t obtained by assigning the label α to the anchor of a: if $a = \langle D, (\lambda x^{(\beta)}.s)\mathtt{L}u\rangle$, then $\mathtt{lift}(t, a, \alpha) = D[(\lambda x^{\alpha}.s)\mathtt{L}u]$, and analogously for the other kinds of steps.*

Notice that $t = D[r]$ and t' being a variant of t imply that $t' = D'[r']$, where D' and r' are variants of D and r respectively. This observation yields the existence of a natural bijection between the sets of steps of two variant terms.

Given $a, b \in \mathcal{RO}(t)$, we define $a[\![b]\!]a'$ as follows. Let $t \xrightarrow{b} u$ and $\mathtt{lift}(t, a, \alpha) \xrightarrow{b_{\mathcal{L}}} u_{\mathcal{L}}$ where $b_{\mathcal{L}}$ is the step which corresponds to b in $\mathtt{lift}(t, a, \alpha)$. Then $a[\![b]\!]a'$ iff $a' \in \mathcal{RO}(u)$, and the anchor of the step corresponding to a' in $u_{\mathcal{L}}$ is labeled with α.

We remark that any variant of t could be used to compute $a[\![b]\!]$: the result is independent *of the variant, considering the natural bijection between steps in variants.*

Embedding

Given $a, b \in \mathcal{RO}(t)$, we define $a \prec_{\mathtt{L}} b$ iff the anchor of a is to the left of the anchor of b (considering t as a string of symbols). Clearly, $\prec_{\mathtt{L}}$ is a total order so that $a \not\prec_{\mathtt{L}} b$ and $a \neq b$ imply $b \prec_{\mathtt{L}} a$.

Observe that the quotient by the graphical equivalence $\sim$ is not considered: if $t \neq u$ and $t \sim u$, then t and u are two different elements of $\mathcal{O}$.

We define other notions associated with steps.

Definition 4.2.2 (Pattern, box, context of a step)**.** *Given a step $a = \langle D, r\rangle$ or $a = \langle D, r, C\rangle$, the **pattern** of a is r; the **box**[5] of a is the subterm of the pattern noted u in the definition of each kind of step; and the **context** of a is D.*

Fig. 4.2 depicts several concepts associated with steps, using as example a $\mathtt{ls}$-step. Notice that the pattern of two different coinitial steps can coincide. This is the case for $\mathtt{ls}$-steps, as in the two steps in the term $(xx)[x/y]$.

$$\langle \underbrace{\;D\;}_{\text{context}}, \overbrace{C[\![\underbrace{\;x\;}_{\text{anchor}}]\!][x/\underbrace{\;u\;}_{\text{box}}]}^{\text{pattern}}, C\rangle$$

Figure 4.2: Notions associated with steps in $\lambda_{\mathtt{lsub}}$

We already noticed, in the definition of the residual relation, that the definition of steps extends naturally to labeled terms.

Notation 4.2.3. *If t is a labeled term, then we write $\mathtt{Red}_{\alpha}(t)$ for the set of steps in t whose anchors are labeled with α.*

[5]When terms are represented as linear logic proof-nets, what we call *box* corresponds exactly to the exponential box.

We give some examples of the preceding definitions. The term

$$t = (\lambda x.x\,[z/\texttt{b}])((wyw)\,[w/\texttt{a}]\,[y/\texttt{c}])$$

is the source term of the following five $\mathfrak{A}_{\mathrm{L}}$ steps:

- One db-step, whose pattern comprises all the term t. This step is described in $\mathfrak{A}_{\mathrm{L}}$ as $\langle \Box, t \rangle$, and its anchor is the x appearing next to the only occurrence of λ.
- One gc-step, corresponding to the substitution $[z/\texttt{b}]$, whose description is $\langle (\lambda x.\Box)((wyw)\,[w/\texttt{a}]\,[y/\texttt{c}]), x\,[z/\texttt{b}] \rangle$, and whose anchor is z.
- Two ls-steps, corresponding to the two occurrences of w in the subterm wyw. The description of the step corresponding of the leftmost occurrence is $\langle (\lambda x.x\,[x/\texttt{b}])(\Box\,[y/\texttt{c}]), (wyw)\,[w/\texttt{a}], \Box yw \rangle$, and its anchor the corresponding occurrence of w. Notice that the pattern of these two steps, namely $(wyw)\,[w/\texttt{a}]$, coincide.
- One ls-step, corresponding to the occurrence of y in the subterm wyw, whose description is $\langle (\lambda x.x\,[x/\texttt{b}])\Box, (wyw)\,[w/\texttt{a}]\,[y/\texttt{c}], (w\Box w)\,[w/\texttt{a}] \rangle$.

The labeling of the anchors of these steps, using different labels for each, yields the variant

$$t' = (\lambda x^{\tau}.x\,[z^{\alpha}/\texttt{b}])((w^{\beta}y^{\gamma}w^{\delta})\,[w/\texttt{a}]\,[y/\texttt{c}])$$

Let a, b, c, d, e be the steps in t labeled with $\alpha, \beta, \gamma, \delta$ and τ respectively in t'. The relative position of the anchors give the embeddings for these steps, namely $e \prec_{\mathrm{L}} a \prec_{\mathrm{L}} b \prec_{\mathrm{L}} c \prec_{\mathrm{L}} d$. In turn, performing a labeled reduction from t' allows to compute their residuals. Consider

$$\begin{array}{rcll} t' & = & (\lambda x^{\tau}.x\,[z^{\alpha}/\texttt{b}])((w^{\beta}y^{\gamma}w^{\delta})\,[w/\texttt{a}]\,[y/\texttt{c}]) & \\ & \to_{\texttt{db}} & x\,[z^{\alpha}/\texttt{b}]\,[x/(w^{\beta}y^{\gamma}w^{\delta})\,[w/\texttt{a}]\,[y/\texttt{c}]] & = \; t'_1 \\ & \to_{\texttt{ls}} & (w^{\beta}y^{\gamma}w^{\delta})\,[w/\texttt{a}]\,[y/\texttt{c}]\,[z^{\alpha}/\texttt{b}]\,[x/(w^{\beta}y^{\gamma}w^{\delta})\,[w/\texttt{a}]\,[y/\texttt{c}]] & = \; t'_2 \end{array}$$

E.g., we observe that b has one residual after e, which is the β-labeled step in t'_1. In turn, this step has two residuals after the ls-step contracted subsequently, i.e. the two β-labeled steps in t'_2.

Two decisions were made in order to keep $\mathfrak{A}_{\mathrm{L}}$ simple. Firstly, *the quotient of the set of terms by the graphical equivalence $\sim$ is not considered*; each term of $\lambda_{\texttt{lsub}}^{\sim}$ is a separate object. Secondly, the embedding relation $\prec_{\mathrm{L}}$ is a *total order*, being larger than what the intuition of embedding (as described in Section 1.3.1) would suggest: $a \prec_{\mathrm{L}} b$ does not imply that a can have some power on b. E.g. in $(x[x/s])(y[y/u])$, we have $a \prec_{\mathrm{L}} b$ for any a and b inside s and u respectively. The converse does hold: whenever a can have some power on b, $a \prec_{\mathrm{L}} b$ is verified.

In the remainder of this section, we prove that $\mathfrak{A}_{\mathrm{L}}$ verifies the initial axioms, FD, SO, and all the embedding axioms.

It is immediate to verify that Self Reduction and Finite Residuals hold.

Lemma 4.2.4 (Ancestor Uniqueness). *Let $b_1[\![a]\!]b'$ and $b_2[\![a]\!]b'$. Then $b_1 = b_2$.*

Proof. Let us define t as the source of a, b_1 and b_2. Let t' be the variant of t which results of lifting it successively w.r.t. each of its redexes, i.e. the anchor of each redex in t carries a unique label in t'. Let $t' \xrightarrow{a} u'$. Independence of the variant used to compute residuals implies that the residuals of all the steps in t can be obtained by looking for the corresponding label in u'. Therefore, $b_1[\![a]\!]b'$ implies that b_1 has, in t', the same label as b' in u'; and $b_2[\![a]\!]b'$ yields that also b_2 has in t' the same label as b' in u'. Hence, b_1 and b_2 have the same label in t'. Consequently, each redex carrying a unique label in t' implies $b_1 = b_2$. □

We organise the proofs of the other axioms in different subsections.

4.2.1 Finite developments

We define the notion of labeled multiplicities, and then use it to define the measure (called *number of potential labeled redexes*) proving finiteness of developments.

Definition 4.2.5. *The number of* ***labeled multiplicities*** *of well-labeled terms is defined as follows:*

$$
\begin{array}{lcll}
\mathtt{LM}_x(z) & := & 0 \text{ (for all } z) & \\
\mathtt{LM}_x(x^\alpha) & := & 1 & \\
\mathtt{LM}_x(y^\gamma) & := & 0 \text{ (for } x \neq y) & \\
\mathtt{LM}_x(\lambda y.t) & := & \mathtt{LM}_x(t) & \\
\mathtt{LM}_x(tu) & := & \mathtt{LM}_x(t) + \mathtt{LM}_x(u) & \text{if } t \in \mathcal{T}_{\mathcal{WL}} \\
\mathtt{LM}_x((\lambda y^\gamma.t)\mathtt{L}u) & := & \mathtt{LM}_x((\lambda y.t)\mathtt{L}) + \mathtt{LM}_x(u) & \\
\mathtt{LM}_x(t[y^\gamma/u]) & := & \mathtt{LM}_x(t) + \mathtt{LM}_x(u) & \\
\mathtt{LM}_x(t[y/u]) & := & \mathtt{LM}_x(t) + \mathtt{LM}_x(u) + \mathtt{LM}_y(t) \cdot \mathtt{LM}_x(u) & \\
\mathtt{LM}_x(t[x/u]) & := & 0 &
\end{array}
$$

Note that $\mathtt{LM}_x(t) = 0$ *if* $x \notin \mathtt{flv}(t)$.

Definition 4.2.6. *The number of* ***potential labeled redexes*** *of well-labeled terms is defined as follows:*

$$
\begin{array}{lcll}
\mathtt{PLR}(x) = \mathtt{PLR}(x^\alpha) & := & 0 & \\
\mathtt{PLR}(\lambda x.t) & := & \mathtt{PLR}(t) & \\
\mathtt{PLR}(tu) & := & \mathtt{PLR}(t) + \mathtt{PLR}(u) & \text{if } t \in \mathcal{T}_{\mathcal{WL}} \\
\mathtt{PLR}((\lambda x^\alpha.v)\mathtt{L}u) & := & 1 + \mathtt{PLR}((\lambda x.v)\mathtt{L}) + \mathtt{PLR}(u) & \\
\mathtt{PLR}(t[x/u]) & := & \mathtt{PLR}(t) + \mathtt{PLR}(u) + \mathtt{LM}_x(t) \cdot \mathtt{PLR}(u) + \mathtt{LM}_x(t) & \\
\mathtt{PLR}(t[x^\alpha/u]) & := & 1 + \mathtt{PLR}(t) + \mathtt{PLR}(u) &
\end{array}
$$

Remark that for every $t \in \mathcal{T}_{\mathcal{WL}}$ *containing at least one redex we have* $\mathtt{PLR}(t) > 0$.

Several lemmas are needed to prove that the PLR measure decreases with each step of labeled reduction. The proof of some of these auxiliary lemmas can be found in Appendix C.2.

Lemma 4.2.7. *Let* $C[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ *and* $t \in \mathcal{T}_{\mathcal{WL}}$ *such that* $\mathtt{fv}(t) \cap \mathtt{bv}(C) = \emptyset$. *Then* $C[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$.

Proof. See Appendix C.1, page 260. □

Lemma 4.2.8. *Let t, x be such that $x \notin \texttt{flv}(t)$. Then $\texttt{LM}_x(t) = 0$.*

Proof. Straightforward induction on t. □

Lemma 4.2.9. *Let $C[\![x^\alpha]\!], u \in \mathcal{T}_{\mathcal{WL}}$ and a variable y such that $x \neq y$, $\texttt{fv}(u) \cap \texttt{bv}(C) = \emptyset$, and $x, y \notin \texttt{fv}(u)$. Then: (i) $\texttt{LM}_x(C[\![x^\alpha]\!]) > \texttt{LM}_x(C[\![u]\!])$, and (ii) $\texttt{LM}_y(C[\![x^\alpha]\!]) = \texttt{LM}_y(C[\![u]\!])$.*

Proof. See Appendix C.2, page 263. □

Lemma 4.2.10. *Let $C[\![y^\gamma]\!] \in \mathcal{T}_{\mathcal{WL}}$, $u \in \mathcal{T}_{\mathcal{WL}}$ and x variable, such that $x \neq y$, $y \notin \texttt{fv}(u)$ and $x \notin \texttt{bv}(C)$. Then $\texttt{LM}_x(C[\![y^\gamma]\!]) + \texttt{LM}_y(C[\![y^\gamma]\!]) \cdot \texttt{LM}_x(u) = \texttt{LM}_x(C[\![u]\!]) + \texttt{LM}_y(C[\![u]\!]) \cdot \texttt{LM}_x(u)$.*

Proof. See Appendix C.2, page 264. □

Lemma 4.2.11. *Let $C[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$ and $u \in \mathcal{T}_{\mathcal{WL}}$ such that $x \notin \texttt{fv}(u)$. Then $\texttt{PLR}(C[\![x^\alpha]\!]) + \texttt{LM}_x(C[\![x^\alpha]\!]) \cdot \texttt{PLR}(u) = \texttt{PLR}(C[\![u]\!]) + \texttt{LM}_x(C[\![u]\!]) \cdot \texttt{PLR}(u)$.*

Proof. See Appendix C.2, page 265. □

Lemma 4.2.12. *Let $t, u \in \mathcal{T}_{\mathcal{WL}}$ such that $t \approx u$, and $x \notin \texttt{bv}(t)$. Then (i) $\texttt{LM}_x(t) = \texttt{LM}_x(u)$, and (ii) $\texttt{PLR}(t) = \texttt{PLR}(u)$*

Proof. By case analysis on the equation used in $t \approx u$.

- Assume $t \approx_{\texttt{CS}} u$. Recall that $z \notin \texttt{fv}(s_2)$ and $y \notin \texttt{fv}(s_3)$.

  If $t = s_1[y^\alpha/s_2][z^\beta/s_3]$ and $u = s_1[z^\beta/s_3][y^\alpha/s_2]$, then $\texttt{LM}_x(t) = \texttt{LM}_x(u) = \texttt{LM}_x(s_1) + \texttt{LM}_x(s_2) + \texttt{LM}_x(s_3)$, and analogously for PLR.

  If $t = s_1[y^\alpha/s_2][z/s_3]$ and $u = s_1[z/s_3][y^\alpha/s_2]$, then $\texttt{LM}_x(t) = \texttt{LM}_x(u) = \texttt{LM}_x(s_1) + \texttt{LM}_x(s_2) + \texttt{LM}_x(s_3) + \texttt{LM}_z(s_1) \cdot \texttt{LM}_x(s_3)$, and $\texttt{PLR}(t) = \texttt{PLR}(u) = 1 + \texttt{PLR}(s_1) + \texttt{PLR}(s_2) + \texttt{PLR}(s_3) + \texttt{LM}_z(s_1) \cdot \texttt{PLR}(s_3) + \texttt{LM}_z(s_1)$.

  If $t = s_1[y/s_2][z/s_3]$ and $u = s_1[z/s_3][y/s_2]$, then $\texttt{LM}_x(t) = \texttt{LM}_x(u) = \texttt{LM}_x(s_1) + \texttt{LM}_x(s_2) + \texttt{LM}_y(s_1) \cdot \texttt{LM}_x(s_2) + \texttt{LM}_x(s_3) + \texttt{LM}_z(s_1) \cdot \texttt{LM}_x(s_3)$, and $\texttt{PLR}(t) = \texttt{PLR}(u) = \texttt{PLR}(s_1) + \texttt{PLR}(s_2) + \texttt{LM}_y(s_1) \cdot \texttt{PLR}(s_2) + \texttt{LM}_y(s_1) + \texttt{PLR}(s_3) + \texttt{LM}_z(s_1) \cdot \texttt{PLR}(s_3) + \texttt{LM}_z(s_1)$.

- Assume $t \approx_{\sigma_1} u$, so that $t = (\lambda y.s_1)[z^{(\alpha)}/s_2]$ and $u = \lambda y.s_1[z^{(\alpha)}/s_2]$, and $y \notin \texttt{fv}(s_2)$. Notice that $t \in \mathcal{T}_{\mathcal{WL}}$ implies that the y is not labeled. It is straightforward to verify the result, whether z is labeled or not.

- Assume $t \approx_{\sigma_2} u$, so that $t = (s_1 s_2)[y^{(\alpha)}/s_3]$, $u = s_1[y^{(\alpha)}/s_3]s_2$, and $y \notin \texttt{fv}(s_2)$. Consider $s_1' = s_1$ if $s_1 \in \mathcal{T}_{\mathcal{WL}}$, and $s_1' = (\lambda z.s_4)\texttt{L}$ if $s_1 = (\lambda z^\beta.s_4)\texttt{L}$.

  If $t = (s_1 s_2)[y^\alpha/s_3]$ and $u = s_1[y^\alpha/s_3]s_2$, and $s_1 \in \mathcal{T}_{\mathcal{WL}}$, then $\texttt{LM}_x(t) = \texttt{LM}_x(u) = \texttt{LM}_x(s_1') + \texttt{LM}_x(s_2) + \texttt{LM}_x(s_3)$, and analogously for PLR.

  If $t = (s_1 s_2)[y/s_3]$ and $u = s_1[y/s_3]s_2$, and $s_1 \in \mathcal{T}_{\mathcal{WL}}$, then $\texttt{LM}_x(t) = \texttt{LM}_x(u) = \texttt{LM}_x(s_1') + \texttt{LM}_x(s_2) + \texttt{LM}_x(s_3) + \texttt{LM}_y(s_1') \cdot \texttt{LM}_x(s_3)$, and $\texttt{PLR}(t) = \texttt{PLR}(u) = k + \texttt{PLR}(s_1') + \texttt{PLR}(s_2) + \texttt{PLR}(s_3) + \texttt{LM}_y(s_1') \cdot \texttt{PLR}(s_3) + \texttt{LM}_y(s_1')$. □

Lemma 4.2.13. *Let $t, u \in \mathcal{T}_{\mathcal{WL}}$ such that $t \sim u$, and $x \notin \mathtt{bv}(t)$. Then (i) $\mathtt{LM}_x(t) = \mathtt{LM}_x(u)$, and (ii) $\mathtt{PLR}(t) = \mathtt{PLR}(u)$*

Proof. By induction on the characterisation of $\sim$ as the reflexive-transitive closure of $\overset{1}{\sim}$. The interesting case is $t \overset{1}{\sim} u$. We verify this case by induction on $|C|$ where $t = C[t']$, $u = C[u']$ and $t' \approx u'$. If $C = \Box$ then we conclude by Lem. 4.2.12.

If $C = C_1 s$ and $C_1[t'] = (\lambda y^\beta.s)\mathtt{L}$, then there are several cases to consider.

- If $C_1 = \Box$, so that $t' = (\lambda y^\beta.s)\mathtt{L}$, then let us define $t'' = (\lambda y.s)\mathtt{L}$. Case analysis yields $u' = (\lambda y^\beta.s')\mathtt{L}'$, and also $t'' \approx u''$ where $u'' = (\lambda y.s')\mathtt{L}'$. Observe that
  $\mathtt{LM}_x(C[t']) = \mathtt{LM}_x(t'') + \mathtt{LM}_x(s)$, $\mathtt{PLR}(C[t']) = 1 + \mathtt{PLR}(t'') + \mathtt{PLR}(s)$,
  $\mathtt{LM}_x(C[u']) = \mathtt{LM}_x(u'') + \mathtt{LM}_x(s)$, $\mathtt{PLR}(C[u']) = 1 + \mathtt{PLR}(u'') + \mathtt{PLR}(s)$.
  Hence Lem. 4.2.12 allows to conclude.
- If $C_1 = \Box\mathtt{L}$, so that $t' = \lambda y^\beta.s$, then let us define $t'' = \lambda y.s$. Case analysis yields $u' = (\lambda y^\beta.s')\mathtt{L}$ and $t'' \approx u''$ where $u'' = (\lambda y.s')\mathtt{L}$. Observe that
  $\mathtt{LM}_x(C[t']) = \mathtt{LM}_x(C_1[t'']) + \mathtt{LM}_x(s)$, $\mathtt{PLR}(C[t']) = 1 + \mathtt{PLR}(C_1[t'']) + \mathtt{PLR}(s)$
  $\mathtt{LM}_x(C[u']) = \mathtt{LM}_x(C_1[u'']) + \mathtt{LM}_x(s)$, $\mathtt{PLR}(C[u']) = 1 + \mathtt{PLR}(C_1[u'']) + \mathtt{PLR}(s)$.
  Hence IH on $C_1[t''] \overset{1}{\sim} C_1[u'']$ allows to conclude.
- If $C_1 = (\lambda y^\beta.C_2)\mathtt{L}$, or $C_1 = (\lambda y^\beta.s)\mathtt{L}_1[z/C_2]\mathtt{L}_2$, then let C_1' be the result of replacing y^β by y in C_1. We have
  $\mathtt{LM}_x(C[t']) = \mathtt{LM}_x(C_1'[t']) + \mathtt{LM}_x(s)$, $\mathtt{PLR}(C[t']) = 1 + \mathtt{PLR}(C_1'[t']) + \mathtt{PLR}(s)$
  and analogously for $C[u']$. Hence IH suffices to conclude.

If $C = C_1[y/s]$, so that $C_1[t'], s \in \mathcal{T}_{\mathcal{WL}}$, then variable convention implies $y \notin \mathtt{fv}(t')$. For *(i)* we have
$\mathtt{LM}_x(C[t']) = \mathtt{LM}_x(C_1[t']) + \mathtt{LM}_x(s) + \mathtt{LM}_y(C_1[t']) \cdot \mathtt{LM}_x(s)$, and
$\mathtt{LM}_x(C[u']) = \mathtt{LM}_x(C_1[u']) + \mathtt{LM}_x(s) + \mathtt{LM}_y(C_1[u']) \cdot \mathtt{LM}_x(s)$. Then applying IH twice, for $\mathtt{LM}_x$ and $\mathtt{LM}_y$, allows to conclude. For *(ii)* we have
$\mathtt{PLR}(C[t']) = \mathtt{PLR}(C_1[t']) + \mathtt{PLR}(s) + \mathtt{LM}_y(C_1[t']) \cdot \mathtt{PLR}(s) + \mathtt{LM}_y(C_1[t'])$, and
$\mathtt{PLR}(C[u']) = \mathtt{PLR}(C_1[u']) + \mathtt{PLR}(s) + \mathtt{LM}_y(C_1[u']) \cdot \mathtt{PLR}(s) + \mathtt{LM}_y(C_1[t'])$. Therefore IH:*(i)* for $\mathtt{LM}_y$ and IH:*(ii)* allow to conclude.

For the remaining cases, a simple inductive argument applies. ☐

Lemma 4.2.14. *Let $t, u \in \mathcal{T}_{\mathcal{WL}}$ such that $t \overset{\alpha}{\mapsto} u$, and $x \notin \mathtt{bv}(t)$. Then (i) $\mathtt{LM}_x(t) \geqslant \mathtt{LM}_x(u)$, and (ii) $\mathtt{PLR}(t) > \mathtt{PLR}(u)$.*

Proof. By case analysis on the used rule.

- If $t = (\lambda y^\alpha.s_1)\mathtt{L}s_2 \overset{\alpha}{\mapsto}_{\mathtt{db}} s_1[y/s_2]\mathtt{L} = u$, then let us define $\mathtt{L} = [x_1/t_1]\ldots[x_n/t_n]$. Variable convention on t implies $y \notin \mathtt{fv}(t_i)$ for all i, $y \notin \mathtt{fv}(s_2)$, and also $x_i \notin \mathtt{fv}(s_2)$ for all i. Therefore $(\lambda y.s_1)\mathtt{L} \sim \lambda y.s_1\mathtt{L}$ and $u = s[y/s_2]\mathtt{L} \sim s_1\mathtt{L}[y/s_2]$. Hence Lem. 4.2.13 implies $\mathtt{LM}_x((\lambda y.s_1)\mathtt{L}) = \mathtt{LM}_x(\lambda y.s_1\mathtt{L})$, $\mathtt{LM}_x(u) = \mathtt{LM}_x(s_1\mathtt{L}[y/s_2])$, and analogously for $\mathtt{PLR}$.

  Moreover, $t \in \mathcal{T}_{\mathcal{WL}}$ implies $\lambda y.s_1 \in \mathcal{T}_{\mathcal{WL}}$, then $y \notin \mathtt{flv}(s_1)$, so that $y \notin \mathtt{fv}(t_i)$ for all i implies $\mathtt{LM}_y(s_1\mathtt{L}) = 0$; cfr. Lem. 4.2.8.

  For *(i)*, is enough to observe

$\texttt{LM}_x(t) = \texttt{LM}_x((\lambda y.s_1)\texttt{L}) + \texttt{LM}_x(s_2) = \texttt{LM}_x(\lambda y.s_1\texttt{L}) + \texttt{LM}_x(s_2) = \texttt{LM}_x(s_1\texttt{L}) + \texttt{LM}_x(s_2)$
and
$\texttt{LM}_x(u) = \texttt{LM}_x(s_1\texttt{L}[y/s_2]) = \texttt{LM}_x(s_1\texttt{L}) + \texttt{LM}_x(s_2) + \texttt{LM}_y(s_1\texttt{L}) \cdot \texttt{LM}_x(s_2) = \texttt{LM}_x(s_1\texttt{L}) + \texttt{LM}_x(s_2)$.

For *(ii)*, it suffices to observe
$\texttt{PLR}(t) = 1 + \texttt{PLR}((\lambda y.s_1)\texttt{L}) + \texttt{PLR}(s_2) = 1 + \texttt{PLR}(\lambda y.s_1\texttt{L}) + \texttt{PLR}(s_2) = 1 + \texttt{PLR}(s_1\texttt{L}) + \texttt{PLR}(s_2)$
and
$\texttt{PLR}(u) = \texttt{PLR}(s_1\texttt{L}[y/s_2]) = \texttt{PLR}(s_1\texttt{L}) + \texttt{PLR}(s_2) + \texttt{LM}_y(s_1\texttt{L}) \cdot \texttt{PLR}(s_2) + \texttt{LM}_y(s_1\texttt{L}) = \texttt{PLR}(s_1\texttt{L}) + \texttt{PLR}(s_2)$.

- If $t = C[\![y^\alpha]\!][y/s] \overset{\alpha}{\mapsto}_{\texttt{ls}} C[\![s]\!][y/s] = u$, then variable convention implies $y \notin \texttt{fv}(s)$.

  For *(i)* we have
  $\texttt{LM}_x(t) = \texttt{LM}_x(C[\![y^\alpha]\!]) + \texttt{LM}_x(s) + \texttt{LM}_y(C[\![y^\alpha]\!]) \cdot \texttt{LM}_x(s)$, and
  $\texttt{LM}_x(u) = \texttt{LM}_x(C[\![s]\!]) + \texttt{LM}_x(s) + \texttt{LM}_y(C[\![s]\!]) \cdot \texttt{LM}_x(s)$
  Therefore Lem. 4.2.10 suffices to conclude.

  For *(ii)* we have
  $\texttt{PLR}(t) = \texttt{PLR}(C[\![y^\alpha]\!]) + \texttt{PLR}(s) + \texttt{LM}_y(C[\![y^\alpha]\!]) \cdot \texttt{PLR}(s) + \texttt{LM}_y(C[\![y^\alpha]\!])$
  and
  $\texttt{PLR}(u) = \texttt{PLR}(C[\![s]\!]) + \texttt{PLR}(s) + \texttt{LM}_y(s) \cdot \texttt{PLR}(s) + \texttt{LM}_y(s)$
  Moreover, Lem. 4.2.11 and Lem. 4.2.9:*(i)* imply $\texttt{PLR}(C[\![y^\alpha]\!]) + \texttt{LM}_y(C[\![y^\alpha]\!]) \cdot \texttt{PLR}(s) = \texttt{PLR}(C[\![s]\!]) + \texttt{LM}_y(s) \cdot \texttt{PLR}(s)$, and $\texttt{LM}_y(C[\![y^\alpha]\!]) > \texttt{LM}_y(s)$ respectively. Thus we conclude.

- If $t = s_1[y^\alpha/s_2] \overset{\alpha}{\mapsto}_{\texttt{gc}} s_1 = u$, then $\texttt{LM}_x(t) = \texttt{LM}_x(s_1) + \texttt{LM}_x(s_2) \geqslant \texttt{LM}_x(s_1) = \texttt{LM}_x(u)$, and $\texttt{PLR}(t) = 1 + \texttt{PLR}(s_1) + \texttt{PLR}(s_2) > \texttt{PLR}(s_1) = \texttt{PLR}(u)$. □

Lemma 4.2.15. *Let $t, u \in \mathcal{T}_{\mathcal{WL}}$ such that $t \overset{\alpha}{\to} u$, and $x \notin \texttt{bv}(t)$. Then (i) $\texttt{LM}_x(t) \geqslant \texttt{LM}_x(u)$, and (ii) $\texttt{PLR}(t) > \texttt{PLR}(u)$.*

Proof. By induction on $|C|$ where $t = C[t']$, $u = C[u']$ and $t' \overset{\alpha}{\mapsto} u'$. If $C = \Box$ then we conclude by Lem. 4.2.14.

For the inductive cases, an argument similar to that described in the proof of Lem. 4.2.13 applies, changing references to $\approx$ by $\overset{\alpha}{\mapsto}$, and the reference to Lem. 4.2.12 by Lem. 4.2.14. The straightforward cases are the same, those detailed admit similar arguments. The case $C = C_1 s$, subcase $C_1 = \Box\texttt{L}$ does not apply to this lemma, because $t' = \lambda y^\beta.s$ implies that there is no u' satisfying $t' \overset{\alpha}{\mapsto} u'$. □

Proposition 4.2.16. *Let $t \in \mathcal{T}_{\mathcal{WL}}$ and let $\mathbb{L}$ be the set of all the labels of the redexes in t. Then the reduction relation $\to_{\mathbb{L}}$ is terminating.*

Proof. Immediate corollary of Lem. 4.2.15. □

Proposition 4.2.17. *The ARS $\mathfrak{A}_{\texttt{L}}$ enjoys the axiom FD.*

Proof. Let t be a term and $\mathcal{A} \subseteq \mathcal{RO}(t)$. We consider the term t' which results of applying successively the lift operation, to assign a label to the anchor of each step in $\mathcal{A}$. Let $\mathbb{L}$ be the set of the labels used to lift t. Given that residuals are defined in $\mathfrak{A}_{\texttt{L}}$ in terms of labels, Prop. 4.2.16 entails the result. □

4.2.2 Semantic orthogonality

The definition of residuals in terms of labels allows to prove the axiom SO, cfr. Section 2.1.4, for $\mathfrak{A}_{\mathtt{L}}$ by a level-based argument.

Let $a, b, c \in \mathcal{RO}(t)$. We lift t to assign the labels α, β and γ to a, b and c respectively, let us call t' the labeled term obtained. A different rewrite relation corresponds to each label, consider the relations $\xrightarrow{\alpha}$ and $\xrightarrow{\beta}$. The term t' includes exactly one $\xrightarrow{\alpha}$-step and exactly one $\xrightarrow{\beta}$-step. Let $t' \xrightarrow{\alpha} t_1$ and $t' \xrightarrow{\beta} t_2$. Observe that a complete development of $b[\![a]\!]$ corresponds to a $\xrightarrow{\beta}$ reduction sequence from t_1 to a $\xrightarrow{\beta}$-normal form, and similarly for $a[\![b]\!]$. Therefore, to prove SO, it suffices to verify that the $\xrightarrow{\beta}$-normal form of t_1 and the $\xrightarrow{\alpha}$-normal form of t_2 coincide as labeled terms. In such case, the (unlabeled) targets of $a; b[\![a]\!]$ and $b; a[\![b]\!]$ coincide, and also the residuals of the coinitial step c, which are exactly the γ-labeled steps in the common labeled target.

In turn, the equality of the normal forms coincides with the local commutativity of the relations $\xrightarrow{\alpha}$ and $\xrightarrow{\beta}$. Assume that $t_1 \overset{\beta}{\twoheadrightarrow} t''$ and $t_2 \overset{\alpha}{\twoheadrightarrow} t'''$. Observe that t_1 does not include α labels, and therefore neither does t''. Analogously, t''' does not include β labels. Hence, $t'' = t'''$ implies that term not to include neither α nor β labels, i.e., to be a $\xrightarrow{\alpha}$- and a $\xrightarrow{\beta}$-normal form.

These considerations motivate the following statement:

Lemma 4.2.18. *The reduction relations $\xrightarrow{\alpha}$ and $\xrightarrow{\beta}$ locally commute, i.e. if $t, u_1, u_2 \in \mathcal{T}_{\mathcal{WL}}$, $t \xrightarrow{\alpha} u_1$ and $t \xrightarrow{\beta} u_2$ then there exists s s.t. $u_1 \overset{\beta}{\twoheadrightarrow} s$ and $u_2 \overset{\alpha}{\twoheadrightarrow} s$.*

Proof. Let a and b the steps contracted in $t \xrightarrow{\alpha} u_1$ and $t \xrightarrow{\beta} u_2$ respectively. Let D_1, r_1 be the context and pattern of a, and D_2, r_2 those of b, so that $t = D_1[r_1] = D_2[r_2]$.

If $a = b$, or there exists a context E verifying $D_1 = E[\Box, r_2]$ and $D_2 = E[r_1, \Box]$, then a straightforward argument allows to conclude. Therefore, we assume wlog that $D_1 = D_2$ or $D_2 = D_1[D']$ for some context D'. Let us define $r_2 \xrightarrow{\beta} r_2'$. We analyse the possible cases on a.

Assume that a is a $\mathtt{ls}$-step, i.e. $r_1 = (\lambda x^\alpha.s)\mathtt{L}u$.

- If r_2 is inside s, i.e. if $D' = (\lambda x^\alpha.E)\mathtt{L}u$, then

$$\begin{array}{ccc} D_1[(\lambda x^\alpha.E[r_2])\mathtt{L}u] & \xrightarrow{\quad\alpha\quad} & D_1[E[r_2][x/u]\mathtt{L}] \\ \beta\downarrow & & \downarrow\beta \\ D_1[(\lambda x^\alpha.E[r_2'])\mathtt{L}u] & \underset{\alpha}{\dashrightarrow} & D_1[E[r_2'][x/u]\mathtt{L}] \end{array}$$

  A brief remark on notation: from now on we omit the outer context D_1, which is common to all the forthcoming diagrams.

- If r_2 is inside $\mathtt{L}$, or if it is inside u, then a similar argument applies.

- If $s = E[\![y^\beta]\!]$ and $\mathtt{L} = \mathtt{L}_1[y/s']\mathtt{L}_2$, then

$$\begin{array}{ccc} (\lambda x^\alpha.E[\![y^\beta]\!])\mathtt{L}u & \xrightarrow{\quad\alpha\quad} & E[\![y^\beta]\!][x/u]\mathtt{L} \\ \beta\downarrow & & \downarrow\beta \\ (\lambda x^\alpha.E[s'])\mathtt{L}u & \underset{\alpha}{\dashrightarrow} & E[s'][x/u]\mathtt{L} \end{array}$$

Assume that a is a $\texttt{ls}$-step, i.e. $r_1 = C[\![x^\alpha]\!]\,[x/u]$.

- If $C = C'[r_2, \Box]$, so that $D' = C'[\Box, x^\alpha]\,[x/u]$, then a simple argument suffices to conclude.
- If x^α is inside r_2, i.e. $D' = C_1\,[x/u]$ and $r_2 = C_2[\![x^\alpha]\!]$, then we have to analyse the rule for b.

  – Assume that b is a $\texttt{db}$-step, i.e. $r_2 = (\lambda y^\beta.s')\texttt{L}u'$.
  If x^α is inside s', that is $C_2 = (\lambda y^\beta.E)\texttt{L}u'$, then

$$\begin{array}{ccc} C_1[(\lambda y^\beta.E[\![x^\alpha]\!])\texttt{L}u']\,[x/u] & \xrightarrow{\alpha} & C_1[(\lambda y^\beta.E[u])\texttt{L}u']\,[x/u] \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\beta} \\ C_1[E[\![x^\alpha]\!]\,[y/u']\texttt{L}]\,[x/u] & \xrightarrow[\alpha]{} & C_1[E[u]\,[y/u']\texttt{L}]\,[x/u] \end{array}$$

  If x^α is inside either $\texttt{L}$ or u', then a similar argument applies.

  – Assume that b is a $\texttt{ls}$-step, so that $r_2 = E[\![y^\beta]\!]\,[y/u']$.
  If x^α is inside $E[\![y^\beta]\!]$, i.e. $C_2 = E'[\Box, y^\beta]\,[y/u']$ where $E = E'[x^\alpha, \Box]$, then

$$\begin{array}{ccc} C_1[E'[x^\alpha, y^\beta]\,[y/u']]\,[x/u] & \xrightarrow{\alpha} & C_1[E'[u, y^\beta]\,[y/u']]\,[x/u] \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\beta} \\ C_1[E'[x^\alpha, u']\,[y/u']]\,[x/u] & \xrightarrow[\alpha]{} & C_1[E'[u, u']\,[y/u']]\,[x/u] \end{array}$$

  If x^α is inside u', that is $C_2 = E[\![y^\beta]\!]\,[y/E']$, then

$$\begin{array}{ccc} C_1[E[\![y^\beta]\!]\,[y/E'[\![x^\alpha]\!]]]\,[x/u] & \xrightarrow{\alpha} & C_1[E[\![y^\beta]\!]\,[y/E'[u]]]\,[x/u] \\ \downarrow{\scriptstyle\beta} & & \\ C_1[E[E'[\![x^\alpha]\!]]\,[y/E'[\![x^\alpha]\!]]]\,[x/u] & & \downarrow{\scriptstyle\beta} \\ \downarrow{\scriptstyle\alpha} & & \\ C_1[E[E'[u]]\,[y/E'[\![x^\alpha]\!]]]\,[x/u] & \xrightarrow[\alpha]{} & C_1[E[E'[u]]\,[y/E'[u]]]\,[x/u] \end{array}$$

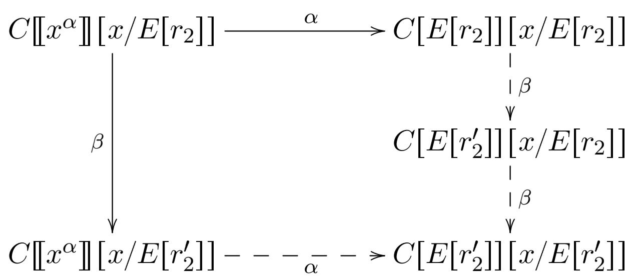
  – Assume that b is a $\texttt{gc}$-step, so that $r_2 = s'\,[y^\beta/u']$.
  If x^α is inside s', i.e. $C_2 = E\,[y^\beta/u']$, then

$$\begin{array}{ccc} C_1[E[\![x^\alpha]\!]\,[y^\beta/u']]\,[x/u] & \xrightarrow{\alpha} & C_1[E[u]\,[y^\beta/u']]\,[x/u] \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\beta} \\ C_1[E[\![x^\alpha]\!]]\,[x/u] & \xrightarrow[\alpha]{} & C_1[E[u]]\,[x/u] \end{array}$$

  If x^α is inside u', that is $C_2 = s'\,[y^\beta/E]$, then

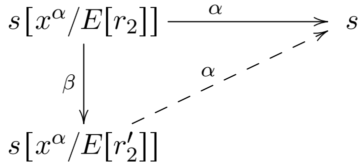
$$\begin{array}{ccc} C_1[s'\,[y^\beta/E[\![x^\alpha]\!]]]\,[x/u] & \xrightarrow{\alpha} & C_1[s'\,[y^\beta/E[u]]]\,[x/u] \\ \downarrow{\scriptstyle\beta} & \swarrow{\scriptstyle\beta} & \\ C_1[s']\,[x/u] & & \end{array}$$

- If $D' = C[\![x^\alpha]\!]\,[x/E]$, then



Assume that a is a gc-step, i.e. $r_1 = s[x^\alpha/u]$.

- If r_2 is inside s, i.e. $D' = E[x^\alpha/u]$, then a simple argument suffices.
- If r_2 is inside u, that is $D' = s[x^\alpha/E]$, then



□

Proposition 4.2.19. *The ARS $\mathfrak{A}_\mathtt{L}$ enjoys the axiom* SO*.*

Proof. Given the argument described at the beginning of this section, Lem. 4.2.18 suffices to conclude: local commutativity of the relations generated by different labels implies semantic orthogonality. □

4.2.3 Embedding axioms

Recall that the embedding relation of $\mathfrak{A}_\mathtt{L}$ is a *total order*. This fact allows to simplify the analysis of the relative embedding of steps, and of its residuals, to a great extent. Moreover, a simple analysis of the rules of $\lambda_{\mathtt{lsub}}$ yields that a step a only has the power to erase, duplicate or change the relative embeddings, on steps whose anchor is on the right of that of a. Therefore, if a affects another step b in any of the described way, then $a <_\mathtt{L} b$. Some examples are given in Fig. 4.3, observe that $a <_\mathtt{L} b$ in all the cases.

These considerations lead to simple proofs of most of the embedding axioms. The exception is Enclave–Creation: a result stating the step creation cases for $\lambda_{\mathtt{lsub}}$ is needed to prove this axiom.

duplication	$x^\alpha[x/(\lambda y^\beta.y)z]$	$\xrightarrow{a}$	$((\lambda y^{\beta'}.y)z)[x/(\lambda y^{\beta''}.y)z]$
erasure	$w[x^\alpha/(\lambda y^\beta.y)z]$	$\xrightarrow{a}$	w
change of relative embeddings			
	$(\lambda w^\alpha.xw)[x/(\lambda y^\beta.y)z](x_1^\gamma[x_1/y_1])$	$\xrightarrow{a}$	$(xw)[w/x_1^{\gamma'}[x_1/y_1]][x/(\lambda y^{\beta'}.y)z]$

a, the α-labeled step, exercises the indicated power over b, the β-labeled one.

Figure 4.3: Different forms of the power of a step over another.

Lemma 4.2.20 (Linearity for $\mathfrak{A}_\mathtt{L}$)**.** *Let $a, b \in \mathcal{RO}(t)$ such that $a \not\leq_\mathtt{L} b$. Then there is exactly one step b' verifying $b[\![a]\!]b'$.*

Proof. By totality of $\prec_{\mathtt{L}}$, we have to show that if $b \prec_{\mathtt{L}} a$ in t then $\exists! b' \;/\; b[\![a]\!]b'$. Now, if a is a db-step this is obvious, as no step is duplicated/erased by a db-step. If a is a $\{\mathtt{gc}, \mathtt{ls}\}$-step then it can only erase or duplicate steps whose anchor is in its box, i.e. on steps on its right, and thus this cannot be the case for b. □

Lemma 4.2.21 (Context-Freeness for $\mathfrak{A}_{\mathtt{L}}$). *Let a, b, c, a', b' be steps such that $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Then the following assertion holds: $a \prec_{\mathtt{L}} c \vee (b \prec_{\mathtt{L}} c \Leftrightarrow b' \prec_{\mathtt{L}} c')$.*

Proof. If $a \not\prec_{\mathtt{L}} c$ then $c \prec_{\mathtt{L}} a$. Assume $b \prec_{\mathtt{L}} c$ (and so $b \prec_{\mathtt{L}} c \prec_{\mathtt{L}} a$). Then, a is on the right of both b and c. It is easily seen that a can only change the order between steps on its right; consequently $b' \prec_{\mathtt{L}} c'$. The other direction is by contraposition. Assume $b \not\prec_{\mathtt{L}} c$, that is $c \prec_{\mathtt{L}} b$. We have to prove that $b' \not\prec_{\mathtt{L}} c'$, i.e. $c' \prec_{\mathtt{L}} b'$. There are two cases. If $c \prec_{\mathtt{L}} b \prec_{\mathtt{L}} a$ then we reason as in the previous direction, getting $c' \prec_{\mathtt{L}} b'$. Otherwise, we have $c \prec_{\mathtt{L}} a \prec_{\mathtt{L}} b$. Now, the only case that is not immediate is when a is a ls-step. It is enough to observe that a ls-step can only move the steps in its box at most where the step itself was; hence, b' can at most be where a was (while the position of c is left unchanged), and so $c' \prec_{\mathtt{L}} b'$. □

Lemma 4.2.22 (Creation lemma for $\lambda_{\mathtt{lsub}}$). *Let $t \xrightarrow{a} t'$, and $b \in \mathcal{RO}(t')$ such that $\varnothing[\![a]\!]b$. Then one of the following conditions holds (where, for readability, β is used to label the created step)*

1. *(db creates a db-step)*
   $t = C[((\lambda x^{\alpha}.(\lambda y.s)\mathtt{L}_1)\mathtt{L}_2\, u)\,\mathtt{L}_3 v] \rightarrow_{\mathtt{db}} C[(\lambda y^{\beta}.s)\mathtt{L}_1[x/u]\mathtt{L}_2\mathtt{L}_3\, v] = t'$
2. *(db creates a ls-step)*
   $t = C[(\lambda x^{\alpha}.D[\![x]\!])\mathtt{L}u] \rightarrow_{\mathtt{db}} C[D[\![x^{\beta}]\!][x/u]\mathtt{L}] = t'$
3. *(db creates a gc-step)*
   $t = C[(\lambda x^{\alpha}.s)\mathtt{L}\, u] \rightarrow_{\mathtt{db}} C[s[x^{\beta}/u]\mathtt{L}] = t'$ *, where $x \notin \mathtt{fv}(s)$*
4. *(ls downward creates a db-step)*
   $t = C[D[x^{\alpha}\mathtt{L}_2\, u][x/(\lambda y.s)\mathtt{L}_1]] \rightarrow_{\mathtt{ls}} C[D[(\lambda y^{\beta}.s)\mathtt{L}_1\mathtt{L}_2 u][x/(\lambda y.s)\mathtt{L}_1]] = t'$
5. *(ls upward creates a db-step)*
   $t = C[x^{\alpha}\mathtt{L}_2\,[x/(\lambda y.s)\mathtt{L}_1]\mathtt{L}_3\, u] \rightarrow_{\mathtt{ls}} C[(\lambda y^{\beta}.s)\mathtt{L}_1\mathtt{L}_2\,[x/(\lambda y.s)\mathtt{L}_1]\mathtt{L}_3 u] = t'$
6. *(ls creates a gc-step)*
   $t = C[D[\![x^{\alpha}]\!][x/u]] \rightarrow_{\mathtt{ls}} C[D[\![u]\!][x^{\beta}/u]] = t'$ *, where $x \notin \mathtt{fv}(D[\![u]\!])$*
7. *(gc creates a gc-step)*
   $t = C[D[s[y^{\alpha}/E[\![x]\!]]][x/u]] \rightarrow_{\mathtt{gc}} C[D[s][x^{\beta}/u]] = t'$ *, where $y \notin \mathtt{fv}(s)$ and $x \notin \mathtt{fv}(D[s])$.*

Proof. See Appendix C.3, page 267. □

Lemma 4.2.23 (Enclave–Creation for $\mathfrak{A}_{\mathtt{L}}$). *Let a, b, b', c' be steps such that $b[\![a]\!]b'$, $\varnothing[\![a]\!]c'$, and $b \prec_{\mathtt{L}} a$. Then $b' \prec_{\mathtt{L}} c'$.*

Proof. A simple inspection of the cases of creation, cfr. Lem. 4.2.22, shows that a step a can create a step c' only on its right or at most where it was, so that c' cannot be on the left of any step that was, in turn, on the left of a. Thus we conclude. □

Lemma 4.2.24 (Enclave–Embedding for $\mathfrak{A}_\mathrm{L}$)**.** *Let a, b, c, b', c' be steps such that $b[\![a]\!]b'$, $c[\![a]\!]c'$, and $b \prec_\mathrm{L} a \prec_\mathrm{L} c$. Then $b' \prec_\mathrm{L} c'$.*

Proof. It suffices to recall that a step can move other steps only up to the point where its anchor was. Therefore the contraction of a cannot provoke the anchor of a residual of c to be on the left of that of b'. □

Lemma 4.2.25 (Stability for $\mathfrak{A}_\mathrm{L}$)**.** *The ARS $\mathfrak{A}_\mathrm{L}$ enjoys the axiom Stability.*

Proof. The hypothesis of the axiom assumes the existence of two steps a and b verifying $a \parallel b$. This case cannot happen when considering $\prec_\mathrm{L}$, which is a total order. Thus we conclude. □

4.3 A first standardisation result

As we have verified in the previous section, the ARS $\mathfrak{A}_\mathrm{L}$ verifies the initial axioms, the axioms FD and SO, and all the embedding axioms. Therefore, our first standardisation result for the linear substitution calculus follows directly from the results for ARS given in [Mel96], and described in Section 2.1.8.

Theorem 4.3.1. *Let γ be a reduction sequence in the $\lambda_{\mathtt{lsub}}$ calculus. Then there exists a* unique *$\mathfrak{A}_\mathrm{L}$-s.r.s. δ such that δ is permutation equivalent to γ.*

Proof. Immediate corollary of Thm. 2.1.24, page 42.

Observe that the embedding $\prec_\mathrm{L}$ is a total order, implying the non-existence of steps a and b verifying $a \neq b$ and $a \parallel b$. Therefore, the relation $\diamond$ generated by $\mathfrak{A}_\mathrm{L}$ coincides with identity. Hence, the existence of a unique δ, instead of just uniqueness of δ modulo $\diamond$, can be stated. □

This standardisation result, while interesting, is not entirely satisfactory, because it does not respect the close relation between $\lambda_{\mathtt{lsub}}$ and proof nets.

The linear substitution calculus has been designed to mimic the representation of λ-calculus in linear logic proof-nets [Gir87], where β-reduction is decomposed into small steps. The relationship between the two formalisms occurs at the static and the dynamic levels: every term can be mapped to a proof-net, and every proof-net can be mapped to a graphical-equivalence class of terms, as this equivalence defined in Section 4.1 by means of the relation $\sim$. Moreover, there is a bijection ϕ between the steps of a term t and the steps of its corresponding proof-net PN_t which induces a *strong bisimulation* between terms and proof-nets: if $t \to_{\lambda_{\mathtt{lsub}}} u$ by reducing a step a, then $PN_t \to_{PN} PN_u$ by reducing $\phi(a)$, and if $PN_t \to_{PN} R$ then there exists a term u s.t. $t \to u$ and $R = PN_u$.

Therefore, one expects that any reasonable notion of standardisation valid in $\lambda_{\mathtt{lsub}}$ can also be applied to proof-nets. There are two reasons which prevent the standardisation result obtained for $\mathfrak{A}_\mathrm{L}$ to meet this requirement.

First, the objects of $\mathfrak{A}_\mathrm{L}$ are not $\sim$-equivalence classes of $\lambda_{\mathtt{lsub}}$ terms, but rather each term is a different object in that model. Consequently, steps and residuals are defined for terms. In order to obtain standardisation results applicable to $\sim$-equivalence classes, we should obtain definitions of steps and residuals which preserve this equivalence. Particularly for *residuals*, notice that if $t \xrightarrow{a} t'$ and b is a step in t, then $b[\![a]\!]$ is a set of steps in t'. It is not immediately clear how to relate these steps in t' with steps in some term t'' verifying $t' \sim t''$.

Second, the total left-to-right order $\prec_{\mathtt{L}}$ is not preserved by the $\sim$-equivalence classes. A simple example follows: consider

$$t' = z[y^\beta/w]r^\alpha \;\sim\; (zr^\alpha)[y^\beta/w] = t''$$

where r^α stands for pattern of a step whose anchor is labeled with α. Let us call a' and a'' the α-labeled step in t' and t'' respectively, and b, b' the $\mathtt{gc}$-steps labeled with β in those terms. We have $b' \prec_{\mathtt{L}} a'$ and $a'' \prec_{\mathtt{L}} b''$, while it is intuitively clear that a'' is the step in t'' corresponding to a in t', and analogously for b'' and b'.

We will address these issues in the following sections.

In Section 4.4 we show that steps and residuals are well-defined w.r.t. $\sim$-equivalent classes, by proving the existence of a bijection between steps in equivalent terms. This bijection preserves residuals: if two steps a and a' in $\sim$-equivalent terms are related by this bijection, and similarly for b and b', then the residuals $b[\![a]\!]$ and $b'[\![a']\!]$ are again related, one-to-one if there are several such residuals.

In Section 4.5, we define a different embedding relation for $\lambda_{\mathtt{lsub}}$, the *box order*, which is preserved by $\sim$-equivalence.

These elements allow to define other ARSs, whose standardisation results are applicable to $\lambda_{\mathtt{lsub}}^{\sim}$, and hence to proof-nets.

4.4 Working with equivalence classes

In this section we first define the notion of residual of a step along a $\sim$-equivalence derivation, so that we will be able to trace steps along $\lambda_{\mathtt{lsub}}^{\sim}$-reduction sequences. We then show that the notions of step and residual defined in Section 4.2 is well-defined w.r.t. equivalence classes, i.e. 1) residuals along equivalence derivations yield a unique bijection between two steps in the same $\sim$-equivalence class, cfr. Lem. 4.4.9, and 2) residuals of rewriting steps lift to $\sim$-equivalence classes, cfr. Lem. 4.4.10.

To trace a step along an equational derivation, we use labels, just like we do to define residuals after $\rightarrow_{\lambda_{\mathtt{lsub}}}$.

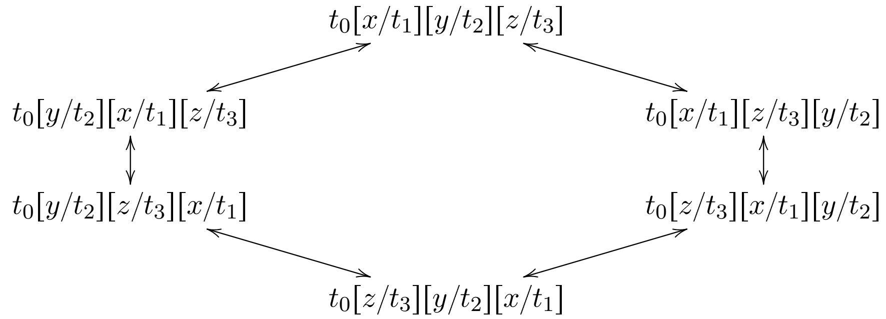
Definition 4.4.1. *Given $t \sim u$, $a \in \mathcal{RO}(t)$ and $\alpha \notin \mathtt{Lab}(t)$, we consider the labeled equation $\mathtt{lift}(t, a, \alpha) \sim u$. The* ***set of residuals of*** *a* ***after*** *$t \sim u$, is given by $a[\![t \sim u]\!] := \{\mathtt{Red}_\alpha(u) \;/\; \mathtt{lift}(t, a, \alpha) \sim u\}$. Again, this definition is* independent *from the variant used to lift the term t. We write $a[\![t \sim u]\!]a'$ iff $a' \in a[\![t \sim u]\!]$ and we extend this notion to sets of steps as expected, in which case we write $\mathcal{A}[\![t \sim u]\!]\mathcal{A}'$, where $\mathcal{A} \subseteq \mathcal{RO}(t)$ and $\mathcal{A}' \subseteq \mathcal{RO}(u)$.*

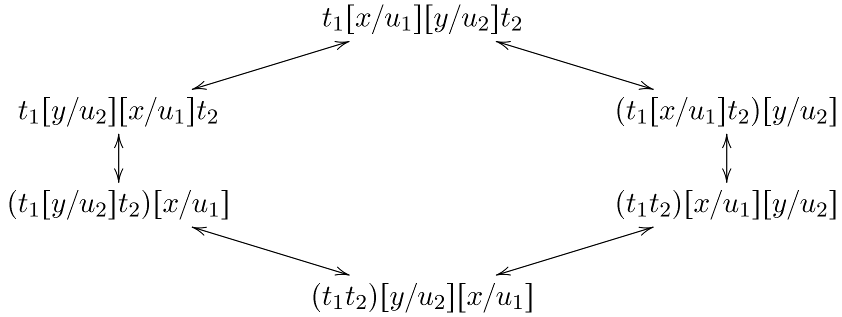
We illustrate this definition with the following example: given $v = (x^\alpha x^\beta)[x/y][x'/y'] \sim (z^\alpha[x'/y']z^\beta)[z/y] = v'$ and $b = \langle \Box[x'/y'], (x^\alpha x^\beta)[x/y], \Box x^\beta \rangle$, we have $b[\![v \sim v']\!] = \{\langle \Box, v', \Box[x'/y']z^\beta \rangle\}$.

Note that our equations do not duplicate/erase/rename labels, so that any step has a unique residual along the equivalence. Therefore, a **bijection** between the steps in $\sim$-equivalent terms can be defined as follows:

Definition 4.4.2. *Let $t \sim t'$. We define the* ***correspondence definitions*** *between $\mathcal{RO}(t)$ and $\mathcal{RO}(t')$, notation $\phi_{t,t'}$, as follows:* $\quad \phi_{t,t'}(a) = a'$ iff $a[\![t \sim t']\!]a'$.

Although this is a quite natural way to relate steps in $\sim$-equivalent terms, its *well-definedness* is not immediate, the reason being the existence of different ways to obtain that two given terms are $\sim$-equivalent. Two examples of this phenomenon follow.





Well-definedness of the bijection between steps could be shown, in principle, by introducing proof-nets and showing that the proof nets corresponding to two $\sim$-equivalent terms s and t are identical: thus there is a bijection between redexes in s and redexes in t, since there is a bijection between redexes in s (resp t) and redexes in their proof-net representation. We prefer, however, to avoid introducing proof-nets here: on the one hand because they are only apparently simpler than terms, and on the other hand, to resort to a unique formalism, namely terms, to develop our ideas.

Consider two unlabeled terms t and t' such that $t \sim t'$, and t_1 a variant of t. For each different $\sim$ derivation justifying $t \sim t'$, there is a corresponding labeled derivation $t_1 \sim t'_1$, where now $\sim$ stands for the labeled graphical equivalence, Dfn. 4.1.5. We must verify that all the possible derivations $t \sim t'$, when lifted to t_1, *attain the same labeled term* t'_1. Otherwise, the well-definedness of the bijection $\phi_{t,t'}$ can be compromised. Consider

$$t = C[(xx)[y/x]][x/u] \sim C[x[y/x]x][x/u] = t'$$

and $t_1 = C[(x^\alpha x^\beta)[y/x^\gamma]][x/u]$, a variant of t. Let a be the step labeled with α in t_1. It is straightforward to verify that

$$t_1 = C[(x^\alpha x^\beta)[y/x^\gamma]][x/u] \sim C[x^\alpha[y/x^\gamma]x^\beta][x/u] = t'_1$$

Suppose that t'_1 were not uniquely determined, e.g. that we could somehow obtain:

$$t_1 = C[(x^\alpha x^\beta)[y/x^\gamma]][x/u] \sim C[x^\beta[y/x^\gamma]x^\alpha][x/u] = t''_1$$

In this case, the definition of $\phi_{t,t'}(a)$ would be ambiguous.

Observe that in such case, transitivity of $\sim$ would entail immediately $C[x^{\alpha}[y/x^{\gamma}]x^{\beta}][x/u] \sim C[x^{\beta}[y/x^{\gamma}]x^{\alpha}][x/u]$. This observation implies that the following statement is a sufficient condition for the well-definedness of $\phi_{t,t'}$:

Let t, t' be labeled terms verifying $t^{\circ} = t'^{\circ}$. Then $t \sim t'$ implies $t = t'$.

To prove this statement, we introduce and verify three structural invariants of labeled steps with respect to the equivalence $\sim$. These invariants are: being a *well-named* term, *substitution address* of each label, and a *partial order* on labels.

Definition 4.4.3. *A term $t \in \mathcal{T}_{\mathcal{WL}}$ is **well-named** iff 1) all its bound variables have pairwise distinct names and 2) all its labels are pairwise distinct.*

The restriction to well-named terms is just given to reason about (the unique occurrence of) each label in a term.

Definition 4.4.4. *For each label α occurring in a well-named term t, we consider the **substitution address** $\mathsf{add}(\alpha, t)$ which is given by the sequence of all the names of the successive substitutions we have to enter in t in order to find α (that is a well-defined and unambiguous sequence because t is well-named).*

For example, if $t = (x^{\alpha}[x/y^{\beta}[z^{\gamma}/y^{\mu}]]y^{\delta})[y/w][x_1^{\nu}/w]$, then the sequence for α and δ is ϵ (i.e. empty), for β is x, for γ and μ is xz, and for ν is x_1.

Definition 4.4.5. *Given a well-named term t, **the order $<_t$ on its labels** is defined as the left-to-right order (looking at t as a string of symbols)* but *only between labels contained in exactly the same substitution, i.e. $\alpha <_t \beta$ iff $\mathsf{add}(\alpha, t) = \mathsf{add}(\beta, t)$ and α appears to the left of β.*

Lemma 4.4.6. *Let t be well-named and $t \sim u$. Then:*

1. *u is well-named and $\mathtt{Lab}(t) = \mathtt{Lab}(u)$.*
2. *$\mathsf{add}(\alpha, t) = \mathsf{add}(\alpha, u)$ for any $\alpha \in \mathtt{Lab}(t)$.*
3. *$<_t = <_u$.*

Proof. Easy induction on the equational derivation $t \sim u$. □

These invariants are used to prove the following lemma, for which we also need to introduce a new concept. Let t and u be s.t. $t^{\circ} = u^{\circ}$. We say that t and u are **equally labeled** if they have labels on exactly the same symbols of t° (but not necessarily the same label).

Lemma 4.4.7. *Let t, u be well-named and equally labeled. If*

1. *$\mathtt{Lab}(t) = \mathtt{Lab}(u)$,*
2. *$\mathsf{add}(\alpha, t) = \mathsf{add}(\alpha, u)$ for any $\alpha \in \mathtt{Lab}(t)$, and*
3. *$<_t = <_u$*

then $t = u$.

Proof. The proof is by induction on the number of labels occurring in t, and then by structure of t.

- If $t = x$ or $t = x^\alpha$ then we conclude immediately.

- Assume $t = \lambda x.t_1$. In this case hypotheses imply $u = \lambda x.u_1$, $t_1{}^\circ = u_1{}^\circ$, $\mathtt{Lab}(t_1) = \mathtt{Lab}(u_1)$, $<_{t_1} = <_{u_1}$, and $\mathtt{add}(\alpha, t_1) = \mathtt{add}(\alpha, u_1)$ for all $\alpha \in \mathtt{Lab}(t_1)$. Then we conclude by the IH.

- Assume $t = t_1 t_2$ and $t_1 \in \mathcal{T}_{\mathcal{WL}}$. In this case hypotheses imply $u = u_1 u_2$, and for $i = 1, 2$, that t_i and u_i are well-named and equally labeled, and also $t_i{}^\circ = u_i{}^\circ$.

  To verify that $\mathtt{Lab}(t_1) = \mathtt{Lab}(u_1)$, assume for contradiction the existence of some $\alpha \in \mathtt{Lab}(t_1) - \mathtt{Lab}(u_1)$. This would imply $\alpha \in \mathtt{Lab}(u_2)$, recall $\mathtt{Lab}(t) = \mathtt{Lab}(u)$. Hypotheses and definition of $\mathtt{add}(\_, \_)$ imply $\mathtt{add}(\alpha, t_1) = \mathtt{add}(\alpha, t) = \mathtt{add}(\alpha, u) = \mathtt{add}(\alpha, u_2)$. In turn, $\mathtt{add}(\alpha, t_1) = \mathtt{add}(\alpha, u_2) = x \cdot k$ would imply both t_1 and u_2 include a substitution for the variable x, contradicting $t^\circ = u^\circ$. Then $\mathtt{add}(\alpha, t_1) = \mathtt{add}(\alpha, u_2) = \epsilon$. On the other hand, t_1 and u_1 being well-named and equally labeled, and $\mathtt{Lab}(t_1) - \mathtt{Lab}(u_1) \neq \varnothing$ would imply the existence of some $\beta \in \mathtt{Lab}(u_1) - \mathtt{Lab}(t_1)$, then $\beta \in \mathtt{Lab}(t_2)$. An argument analogous to that used for α entails $\mathtt{add}(\beta, t_2) = \mathtt{add}(\beta, u_1) = \epsilon$. But then $\alpha <_t \beta$ and $\beta <_u \alpha$, contradicting $<_t = <_u$.

  The existence of some $\alpha \in \mathtt{Lab}(u_1) - \mathtt{Lab}(t_1)$ can be contradicted by a similar argument. Consequently $\mathtt{Lab}(t_1) = \mathtt{Lab}(u_1)$. Therefore, a simple argument on the sets of labels entails $\mathtt{Lab}(t_2) = \mathtt{Lab}(u_2)$. Hence, observing the definitions of substitution addresses and the order on labels is enough to obtain $<_{t_i} = <_{u_i}$ and $\mathtt{add}(\alpha, t_i) = \mathtt{add}(\alpha, u_i)$ for all $\alpha \in \mathtt{Lab}(t_i)$, for $i = 1, 2$, given $<_t = <_u$ and $\mathtt{add}(\alpha, t) = \mathtt{add}(\alpha, u)$ for all $\alpha \in \mathtt{Lab}(t)$. Thus we conclude by applying the IH twice.

- Assume $t = (\lambda x^\alpha.t_1)\mathtt{L}t_2$. In this case hypothesis imply $u = (\lambda x^\beta.u_1)\mathtt{L}'u_2$ (recall t and u are equally labeled). Assuming $\alpha \neq \beta$ would imply $\alpha \in \mathtt{Lab}((\lambda x.u_1)\mathtt{L}'u_2)$, $\beta \in \mathtt{Lab}((\lambda x.t_1)\mathtt{L}t_2)$, and $\mathtt{add}(\alpha, u) = \mathtt{add}(\beta, t) = \epsilon$, then $\alpha <_t \beta$ and $\beta <_u \alpha$, contradicting $<_t = <_u$. Then $\alpha = \beta$, implying $\mathtt{Lab}((\lambda x.t_1)\mathtt{L}t_2) = \mathtt{Lab}((\lambda x.u_1)\mathtt{L}'u_2)$. Moreover, it is straightforward to verify that $(\lambda x.t_1)\mathtt{L}t_2$ and $(\lambda x.u_1)\mathtt{L}'u_2$ verify the remaining hypotheses. Then the IH entails $(\lambda x.t_1)\mathtt{L}t_2 = (\lambda x.u_1)\mathtt{L}'u_2$, thus we conclude.

- If $t = t_1[x/t_2]$, then hypotheses imply that $u = u_1[x/u_2]$. The existence of some $\alpha \in \mathtt{Lab}(t_1) - \mathtt{Lab}(u_1)$, then $\alpha \in \mathtt{Lab}(u_2)$, would imply $\mathtt{add}(\alpha, t_1) \neq \mathtt{add}(\alpha, u_2)$, since t_1 does not include a substitution for the variable x. A similar argument entails $\mathtt{Lab}(u_1) - \mathtt{Lab}(t_1) = \varnothing$, and consequently $\mathtt{Lab}(t_1) = \mathtt{Lab}(u_1)$. In turn, a simple argument on sets of labels imply $\mathtt{Lab}(t_2) = \mathtt{Lab}(u_2)$. It is straightforward to obtain that t_i and u_i verify the remaining hypotheses for $i = 1, 2$. Thus we conclude by applying the IH twice.

- If $t = t_1[x^\alpha/t_2]$. then hypotheses imply $u = u_1[x^\beta/u_2]$. We observe that $\mathtt{add}(\alpha, t) = \mathtt{add}(\beta, u) = x$, then hypotheses entail $\mathtt{add}(\beta, t) = \mathtt{add}(\alpha, u) = x$. Then, assuming $\alpha \neq \beta$ would imply $\alpha \in \mathtt{Lab}(u_2)$, $\mathtt{add}(\alpha, u_2) = \epsilon$, $\beta \in \mathtt{Lab}(t_2)$, and $\mathtt{add}(\beta, t_2) = \epsilon$. But then $\alpha <_t \beta$ and $\beta <_u \alpha$, contradicting $<_t = <_u$. Consequently, $\alpha = \beta$, implying $\mathtt{Lab}(t_1[x/t_2]) = \mathtt{Lab}(u_1[x/u_2])$. Moreover, it is straightforward to verify that $t_1[x/t_2]$ and $u_1[x/u_2]$ verify the remaining hypotheses. Then the IH entails $t_1[x/t_2] = u_1[x/u_2]$, thus we conclude.

$\square$

The preceding lemmas allow to prove the desired condition on $\sim$-equivalent terms, and hence the well-definedness of $\phi_{t,u}$.

Lemma 4.4.8. *Let t be a well-named term having labels exactly at the anchors of all its steps. Let u such that $t \sim u$ and $t^\circ = u^\circ$. Then $t = u$.*

Proof. A simple induction on the equivalence derivation implies that exactly the anchors of the steps in u are labeled, therefore t and u are equally labeled; recall $t^\circ = u^\circ$. Moreover, Lem. 4.4.6:(1) implies that u is well-named, and the whole of Lem. 4.4.6 implies that the remaining requirements of Lem. 4.4.7. Hence we conclude by Lem. 4.4.7. □

Lemma 4.4.9. *Let $t, u \in \mathcal{T}_{\mathcal{WL}}$ such that $t \sim u$. Then $\phi_{t,u}$ is well-defined.*

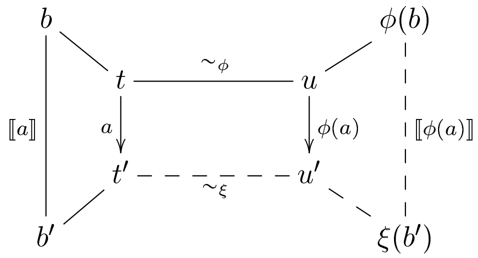
Proof. Assume (by α-conversion) that bound variables in t have pairwise distinct names and consider t_1, the lift of t w.r.t. the (anchors of the) full set of steps $\mathcal{RO}(t)$, so that t_1 is well-named. Consider u_1 such that $t_1 \sim u_1$. Then a simple induction on the equivalence derivation implies that exactly the anchors of the steps in u are labeled, implying the existence of exactly one residual of each step in t (which is uniquely labeled in t_1). Moreover, Lem. 4.4.8 implies that u_1 is unique for $t_1 \sim u_1$ and $u_1{}^\circ = u$. Thus we conclude. □

Lem. 4.4.9 allows to write $t \sim_\phi u$ to denote $\phi_{t,u}$: ϕ is a bijection uniquely determined between the steps of t and of u. We prove that this bijection preserves targets and residuals.

Lemma 4.4.10. *Let $t \sim_\phi u$. Consider $a, b \in \mathcal{RO}(t)$. If $t \xrightarrow{a} t'$, then:*

1. ***Simulation:** $u \xrightarrow{\phi(a)} u'$ with*
2. ***Same equivalence target:** $t' \sim u'$, i.e. $\exists \xi$ s.t. $t' \sim_\xi u'$, and*
3. ***Preservation of residuals:** if $b[\![a]\!]b'$, then $\phi(b)[\![\phi(a)]\!]\xi(b')$.*

Proof. The following diagram depicts the statement



We proceed by induction on the derivation $t \sim u$, considered as a sequence of applications of one equation in either way, inside a context. We consider the case when t and u are one equation application away, i.e. $t = E[t_1]$, $u = E[u_1]$ and $t_1 \approx u_1$.

In the following, we consider that the steps a and b are labeled in t by α and β respectively. Let us call D' and r the context and pattern of a respectively, and r' the term verifying $t' = D'[r']$. We analyse some cases for which the result can be established independently of the nature of t_1 and u_1.

Assume that $D' = D[\Box, t_1]$, so that $E = D[r, \Box]$. Therefore

$$\begin{array}{ccc} t = D[r, t_1] & \overset{\sim_\phi}{\longrightarrow} & D[r, u_1] = u \\ a \downarrow & & \downarrow \phi(a) \\ t' = D[r', t_1] & \underset{\sim_\xi}{- - - -} & D[r', u_1] = u' \end{array}$$

Items 1 and 2 hold immediately.

If the anchor of b is inside D, then $b[\![a]\!] = \{b'\}$ where b' is the β-labeled step in t', and $\phi(b)$ is the β-labeled step in u. In turn, $\xi(b')$ is the β-labeled step in u'. Therefore, it is straightforward to verify that $\phi(b)[\![\phi(a)]\!] = \{\xi(b')\}$. Hence item 3 holds for b.

If the anchor of b is inside r, then $b[\![a]\!]$ is the set of β-labeled steps in r' inside t', and $\phi(b)$ is the β-labeled step inside r in u. Therefore $\phi(b)[\![\phi(a)]\!]$ is the set of α-labeled steps in r' inside u', where the labeled variants of r' inside t' and u' coincide. On the other hand, for any $b' \in b[\![a]\!]$, $\xi(b')$ is a β-labeled step in r' inside u'. Hence item 3 holds for b.

We observe that in any case, item 3 yield from an analysis similar to those just described. Therefore we will check only items 1 and 2 in the following. In all cases justifying diagrams, the result is immediate from the diagram, so we will not remark this fact after each one.

Assume that a is a $\mathtt{ls}$-step, so that $t = D'[C[\![x^\alpha]\!]\,[x/s]]$, and that t_1 is inside s. Then we have

$$\begin{array}{ccc} D'[C[\![x^\alpha]\!]\,[x/s]] & \overset{\sim}{\longrightarrow} & D'[C[\![x^\alpha]\!]\,[x/s']] \\ a \downarrow & & \downarrow \phi(a) \\ D'[C[\![s]\!]\,[x/s]] & \underset{\sim}{- - - -} & D'[C[\![s']\!]\,[x/s']] \end{array}$$

A similar analysis applies if t_1 is inside $C[\![x^\alpha]\!]$, so that $C[\![x^\alpha]\!] \sim C'[\![x^\alpha]\!]$, here it is crucial to observe that $C'[\![x^\alpha]\!]$ includes exactly one α-labeled occurrence of x. An analogous reasoning applies also in analogous cases if a is a $\mathtt{db}$-step or a $\mathtt{gc}$-step.

We analyse the remaining cases.

- Assume that $t_1 \approx_{\mathtt{CS}} t_2$, i.e. $t = E[s_1\,[x/s_2]\,[y/s_3]]$ and $u = E[s_1\,[y/s_3]\,[x/s_2]]$.

  If a is inside some s_i, then a straightforward argument suffices.

  If a is a $\mathtt{ls}$-step on an occurrence of x inside s_1, then we have

$$\begin{array}{ccc} E[C[\![x^\alpha]\!]\,[x/s_2]\,[y/s_3]] & \overset{\sim}{\longrightarrow} & E[C[\![x^\alpha]\!]\,[y/s_3]\,[x/s_2]] \\ a \downarrow & & \downarrow \phi(a) \\ E[C[\![s_2]\!]\,[x/s_2]\,[y/s_3]] & \underset{\sim}{- - - -} & E[C[\![s_2]\!]\,[y/s_3]\,[x/s_2]] \end{array}$$

  If a is a is a $\mathtt{ls}$-step on an occurrence of y inside s_1, then a similar analysis applies. There are no other internal $\mathtt{ls}$-steps, since $y \notin \mathtt{fv}(s_2)$ and $x \notin \mathtt{fv}(s_3)$.

  If a is a $\mathtt{gc}$-step on x, then we have

$$\begin{array}{ccc} E[s_1\,[x^\alpha/s_2]\,[y/s_3]] & \overset{\sim}{\longrightarrow} & E[s_1\,[y/s_3]\,[x^\alpha/s_2]] \\ a \downarrow & & \downarrow \phi(a) \\ E[s_1\,[y/s_3]] & \underset{\equiv}{- - - -} & E[s_1\,[y/s_3]] \end{array}$$

  If a is a $\mathtt{gc}$-step on y, then a similar analysis applies.

  If $E = E'[\Box \mathtt{L} s_4]$ and $s_1 = \lambda z^\alpha.s_1'$, then we have

$$\begin{array}{ccc} E'[(\lambda z^{\alpha}.s'_1)[x/s_2][y/s_3]\mathtt{L}s_4] & \overset{\sim}{\longrightarrow} & E'[(\lambda z^{\alpha}.s'_1)[y/s_3][x/s_2]\mathtt{L}s_4] \\ \downarrow a & & \downarrow \phi(a) \\ E'[s'_1[z/s_4][x/s_2][y/s_3]\mathtt{L}] & \underset{\sim}{\dashrightarrow} & E'[s'_1[z/s_4][y/s_3][x/s_2]\mathtt{L}] \end{array}$$

- Assume that $t_1 \approx_{\sigma_1} t_2$, i.e. $t = E[(\lambda y.s_1)[x/s_2]]$, $u = E[\lambda y.s_1[x/s_2]]$, and $y \notin \mathtt{fv}(s_2)$.

  If a is inside some s_i, then a straightforward argument suffices.

  If a is a $\mathtt{ls}$-step on an occurrence of x in s_1, or it is a $\mathtt{gc}$-step on x, then diagrams similar to those shown for the $\approx_{\mathtt{CS}}$ case can be built.

  If $E = E'[\Box\mathtt{L}s_3]$ and $r = (\lambda y^{\alpha}.s_1)[x/s_2]\mathtt{L}s_3$, then we have

$$\begin{array}{ccc} E'[(\lambda y^{\alpha}.s_1)[x/s_2]\mathtt{L}s_3] & \overset{\sim}{\longrightarrow} & E'[(\lambda y^{\alpha}.s_1[x/s_2])\mathtt{L}s_3] \\ \downarrow a & & \downarrow \phi(a) \\ E'[s_1[y/s_3][x/s_2]\mathtt{L}s_3] & \underset{\sim}{\dashrightarrow} & E'[s_1[x/s_2][y/s_3]\mathtt{L}s_3] \end{array}$$

  Notice that we can assume $x \notin \mathtt{fv}(s_3)$ by variable convention, and we have $y \notin \mathtt{fv}(s_2)$, hence $s_1[y/s_3][x/s_2] \approx_{\mathtt{CS}} s_1[x/s_2][y/s_3]$. This diagram applies only if read right-to-left.

- Assume that $t_1 \approx_{\sigma_2} t_2$, i.e. $t = E[(s_1 s_2)[x/s_3]]$, $u = E[s_1[x/s_3]\,s_2]$, and $x \notin \mathtt{fv}(s_2)$.

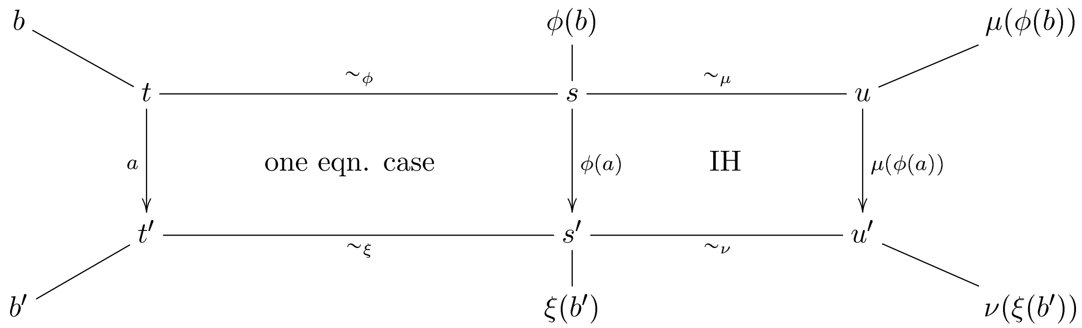
  If a is inside some s_i, then a straightforward argument suffices.

  If a is a $\mathtt{ls}$-step on an occurrence of x in s_1, or it is a $\mathtt{gc}$-step on x, then diagrams similar to those shown for the $\approx_{\mathtt{CS}}$ case can be built.

  If $s_1 = (\lambda y^{\alpha}.s'_1)\mathtt{L}$, then we have

$$\begin{array}{ccc} E[((\lambda y^{\alpha}.s'_1)\mathtt{L}s_2)[x/s_3]] & \overset{\sim}{\longrightarrow} & E[(\lambda y^{\alpha}.s'_1)\mathtt{L}[x/s_3]\,s_2] \\ \downarrow a & & \downarrow \phi(a) \\ E[s'_1[y/s_2]\mathtt{L}[x/s_3]] & \underset{=}{\dashrightarrow} & E[s'_1[y/s_2]\mathtt{L}[x/s_3]] \end{array}$$

We get back to the induction on the derivation $t \sim u$. If $t = u$ then we conclude immediately; the bijections ψ and ξ are the identity on $\mathcal{RO}(t)$ and $\mathcal{RO}(t')$ respectively. Otherwise, consider the following diagram:



Let θ such that $t \sim_{\theta} u$. Uniqueness of θ, cfr. Lem. 4.4.9, implies that $\theta = \mu \cdot \phi$. Then item 1 is immediate, and item 2 also: considering $\psi = \nu \cdot \xi$, we obtain $t' \sim_{\psi} u'$. Item 3 follows from the one-equation case and the IH, given the definitions of θ and ψ. $\Box$

We conclude this section by noticing that we have just shown that the graphical equivalence $\sim$ is a *strong bisimulation* between $\lambda_{\mathtt{lsub}}$ and itself: $t \sim u \to s$ implies the existence of some r verifying $t \to r \sim s$. Moreover, this bisimulation induces a bijection of steps, so it is possible to mimic reduction sequences via $\sim$ as follows: given $t \overset{\delta}{\twoheadrightarrow} u$ and $t \sim t'$, we can unambiguously refer to the simulation $t' \overset{\delta'}{\twoheadrightarrow} u'$ of δ, where $u \sim u'$. This simulation also preserves residuals. Thus we can say that the reduction sequences δ and δ' essentially contract the same sequence of steps.

4.5 The box order on steps

Let us recall the drawbacks of the left-to-right embedding $\prec_{\mathtt{L}}$ defined in Section 4.2, page 102.

Firstly, it does not correspond to the idea of the embedding relation in the ARS model: if $<$ is the embedding for an ARS, then $a < b$ should imply that the step a possibly has some power over the step b, e.g. to erase or duplicate it. In the term

$$x\, s_1\, s_2$$

which is in fact a head normal form, it is clear that no step inside s_1 can have any power over a step inside s_2, and yet $a \prec_{\mathtt{L}} b$ if a and b are inside s_1 and s_2 respectively. Another example is

$$(yx)\,[x/s_1]\,[y/s_2]$$

where again, $a \prec_{\mathtt{L}} b$ if a and b are inside s_1 and s_2 respectively. In other words, $\prec_{\mathtt{L}}$ *captures more* than what the "power principle" suggests.

Secondly, the order $\prec_{\mathtt{L}}$ is not preserved by the graphical equivalence. E.g., we have

$$t = (yx)\,[x/s_1]\,[y/s_2] \sim (yx)\,[y/s_2]\,[x/s_1] = t'$$

where the relative $\prec_{\mathtt{L}}$-embedding of two steps, one inside s_1 and another inside s_2, is different in t than in t'. Another example is described in page 113.

Then we introduce the **box order** $\prec_{\mathtt{B}}$, designed to overcome both shortcomings of $\prec_{\mathtt{L}}$. It is based on the "power principle": if a step a can erase or duplicate a step b, we enforce $a \prec_{\mathtt{B}} b$. Several examples are given in Figure 4.4. In all the cases shown, the downward a, whose anchor is labeled with an α, must precede the rightward b, whose anchor is inside s for Fig. 4.4.a and Fig. 4.4.b, and it is labeled with a β in the remaining two diagrams.

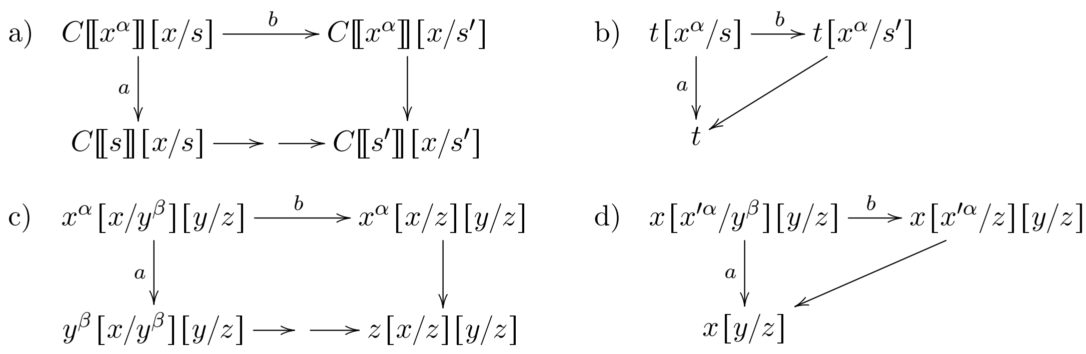


Figure 4.4: Some standardisation diagrams for $\lambda_{\mathtt{lsub}}$.

Observe that the figure actually shows *local confluence* diagrams. A simple diagrammatic intuition, due to [Klo80] and then explored in [Mel05], indicates that whenever a step is duplicated (resp. erased), then the standard reduction sequence should be the longest (resp. shortest) side of the diagram. In all the diagrams, the standard reduction sequence is that going down and then right. This intuition, and also the Linearity axiom, are in line with the "power principle" we use to define $\prec_{\mathtt{B}}$.

Observe that in Fig. 4.4.a and Fig. 4.4.b, the pattern of a syntactically nests that of b; more precisely, it is the *box* of a what nests the pattern of b (cfr. the definition of the box of a step in page 102). This is not the case for Fig. 4.4.c and Fig. 4.4.d. In these cases, we have actually the pattern of b syntactically nesting that of a. Therefore, the intuition indicating the coherence between semantic embedding and syntactic nesting is not valid for the linear substitution calculus.

However, there is a syntactic indication common to all the examples in Fig. 4.4: the *anchor* of the step b is inside the box of the step a. This observation allows to formalise the definition of $\prec_{\mathtt{B}}$.

The formal definition of the **box order** $\prec_{\mathtt{B}}$ follows.

Definition 4.5.1. *Let $a, b \in \mathcal{RO}(t)$. Then,*

- *a **immediately boxes** b, noted $a \prec^1_{\mathtt{B}} b$, if the anchor of b is in the box of a, i.e. if the pattern of a is any of $(\lambda x.t)\mathtt{L}u$, $C[\![x]\!][x/u]$ or $t[x/u]$, then the anchor of b appears in u.*
- *a **boxes** b, noted $a \prec_{\mathtt{B}} b$ if $a(\prec^1_{\mathtt{B}})^+ b$ (we use $a \preceq_{\mathtt{B}} b$ for $a(\prec^1_{\mathtt{B}})^* b$);*

Observe that $\prec_{\mathtt{B}}$ allows embedding to occur *at a distance*. Consider the term $(x^\alpha z[x/y^\beta]yz)[y/z]$. The α-labeled step $\prec_{\mathtt{B}}$-embeds the β-labeled one, while the substitution corresponding to the latter is distant from the pattern of the former.

Notice also that $a \parallel b$ (i.e. $a \neq b$, $a \not\prec_{\mathtt{B}} b$ and $b \not\prec_{\mathtt{B}} a$) does not imply that a and b are syntactically disjoint. Examples: the steps labeled with α and β are disjoint but 1) syntactically superposed in $(x^\alpha\, x^\beta)[x/y]$, and 2) syntactically nested in $(\lambda z^\alpha.x^\beta[x/z])y$. However, disjoint steps always strongly locally commute in the following sense: if $t_0 \xrightarrow{\alpha} t_1$ and $t_0 \xrightarrow{\beta} t_2$ then there exists t_3 s.t. $t_1 \xrightarrow{\beta} t_3$ and $t_2 \xrightarrow{\alpha} t_3$. Note that this is just a particular case of SO where the diagram can be closed by using just one reduction step from t_i to t_3. This observation shows the semantic adequacy of the box order.

A final remark about $\prec_{\mathtt{B}}$: the relation $\prec_{\mathtt{B}}$ is not contained in $\prec^1_{\mathtt{B}}$, therefore the definition of the former is not redundant. This phenomenon is caused by chains of $\mathtt{ls}$-steps where the anchor of each one is inside the box of the following, as in the term $x^\alpha y[x/y^\beta][y/z^\gamma][z/x'y']$. If we call a, b and c the steps labeled with α, β and γ respectively, then we have $a \prec^1_{\mathtt{B}} b \prec^1_{\mathtt{B}} c$, implying $a \prec_{\mathtt{B}} c$, but not $a \prec^1_{\mathtt{B}} c$.

The box order preserves the graphical equivalence $\sim$, thus solving the second shortcoming mentioned at the beginning of this section. For example, for $t[x/u][y/v]$ with $y \notin \mathtt{fv}(u)$ the redexes in u and the redexes in v are not related by $\prec_{\mathtt{B}}$, so that $\prec_{\mathtt{B}}$ is stable by the permuting axiom $t[y/v][x/u] \sim_{\mathtt{CS}} t[x/u][y/v]$ (where $y \notin \mathtt{fv}(u)$ & $x \notin \mathtt{fv}(v)$). More precisely, given $s \sim t$, the bijection between $\mathcal{RO}(s)$ and $\mathcal{RO}(t)$ defined in Section 4.4 is order-preserving. Formally

Lemma 4.5.2. *Let t, u be terms s.t. $t \sim_\phi u$, where ϕ is the bijection described in page 114, cfr. Lem. 4.4.9. Then, ϕ commutes with $\prec_{\mathtt{B}}$, i.e. $a \prec_{\mathtt{B}} b$ iff $\phi(a) \prec_{\mathtt{B}} \phi(b)$.*

Proof. It suffices to remark that symbols cannot go in/outside the box of a step, including those of db-steps, by means of the $\sim$ relation. See Appendix C.4, page 271, for the technical details. □

Consequently, $<_{\mathtt{B}}$ can be thought as a relation on $\sim$-equivalence classes. This result, along with those described in Section 4.4, cfr. Lem. 4.4.10, implies that the definition of $\lambda_{\mathtt{lsub}}^{\sim}$ as a rewriting system on $\sim$-equivalence classes behaves as expected, despite the reduction relation being defined on terms. That is: the set of steps of a $\sim$-equivalence class, the target and residuals of a step, and how the steps are related by $<_{\mathtt{B}}$, do not depend on the term used to compute them.

This observation leads to the possibility of defining an ARS to model $\lambda_{\mathtt{lsub}}^{\sim}$, having the box order $<_{\mathtt{B}}$ as the embedding relation. For technical reasons to be discussed later, we actually define *two* ARS.

4.5.1 ARS based on the box order

The box order $<_{\mathtt{B}}$ leads to the definition of two ARS, for $\lambda_{\mathtt{lsub}}$ and $\lambda_{\mathtt{lsub}}^{\sim}$ respectively.

Definition 4.5.3. *We define the ARS $\mathfrak{A}_{\mathtt{B}}$ as follows: the objects, steps, source and target functions, and residual relation, are as defined for $\mathfrak{A}_{\mathtt{L}}$, cfr. page 101. The embedding relation is the box order $<_{\mathtt{B}}$, considered as a relation on terms.*

Definition 4.5.4. *We define the ARS $\mathfrak{A}_{\mathtt{B}}^{\sim}$ as follows:*

- *the objects are the $\sim$-equivalence classes of the set of terms of $\lambda_{\mathtt{lsub}}$, i.e., the objects being rewritten in $\lambda_{\mathtt{lsub}}^{\sim}$.*
- *the steps, source, target and residuals are the quotient, by $\sim$-equivalence, of those defined for $\mathfrak{A}_{\mathtt{L}}$, given the bijection between steps in $\sim$-equivalent terms defined in page 114. Lem. 4.4.10 implies that these elements are well-defined.*
- *the embedding relation is the quotient of the box order $<_{\mathtt{B}}$ by $\sim$-equivalence; Lem. 4.5.2 implies its well-definedness.*

We verify that $\mathfrak{A}_{\mathtt{B}}$ and $\mathfrak{A}_{\mathtt{B}}^{\sim}$ enjoy the initial axioms, FD, SO, Linearity and Context-Freeness. The following definition allows to express more concisely, in the following proofs, the possible locations of the pattern of a step within a term.

Definition 4.5.5. *If $t = C[s]$, then we say that **a step** a **in** t **is inside** s, notation $a \subseteq s$, iff its context is $D = C[D_1]$, so that its pattern is a subterm of s. We define analogously the meaning of a step a being inside a substitution list* L*, and denote $a \subseteq \mathtt{L}$. We write $a \bowtie s$ iff the pattern of a is exactly the displayed occurrence of s, so that its context is C. Notice that $a \bowtie s$ implies $a \subseteq s$.*

We focus on $\mathfrak{A}_{\mathtt{B}}$ first. The proofs of the initial axioms, FD and SO given for $\mathfrak{A}_{\mathtt{L}}$, are immediately valid for $\mathfrak{A}_{\mathtt{B}}$, since these axioms are not related with the embedding relation of an ARS.

The proof of Linearity follows; it is preceded by an auxiliary lemma.

Lemma 4.5.6. *Let $t = C[\![x^{\beta}]\!]$ where the indicated is only the occurrence of the label β in t, and $t \xrightarrow{a} u$, such that x^{β} is not in the box of a. Then u contains exactly one occurrence of x^{β}.*

Proof. By induction on the context of a. We label the anchor of a using α. For the base case, namely $a \bowtie t$, we consider each rule.

- If $t = (\lambda y^{\alpha}.s_1)\mathtt{L}s_2$, so that $u = s_1[y/s_2]\mathtt{L}$, then the result holds for any possible location for x^{β}.
- If $t = D[y^{\alpha}][y/s_2]$, so that $u = D[s_2][y/s_2]$, then x^{β} not being in the box of a implies that it is inside D, hence we conclude immediately.
- If $t = s_1[y^{\alpha}/s_2]$, a similar, yet simpler, analysis applies.

If $a \not\bowtie t$, then $t = \lambda y.t_1$, $t = t_1t_2$ or $t = t_1[y/t_2]$, and $a \subseteq t_i$ for some i.

If the occurrence of x^{β} lies inside t_i, then we conclude by IH. If this occurrence is inside a different subterm, then we conclude immediately. □

Proposition 4.5.7 (The ARS $\mathfrak{A}_{\mathtt{B}}$ enjoys the axiom Linearity). *Let a, b be two coinitial steps, such that $a \not\leq_{\mathtt{B}} b$. Then $!\exists b' \;/\; b[\![a]\!]b'$.*

Proof. By induction on $n = min(n_a, n_b)$, where n_a is the length, defined as number of symbols, of the context of a, and similarly, n_b is the length of the context of b. We label the anchors of a and b using α and β respectively, and define $t \xrightarrow{a} u$.

The base case is when $n = 0$. We analyse the rules of $\to_{\lambda_{\mathtt{lsub}}}$.

- If $t = (\lambda x^{\alpha}.s_1)\mathtt{L}s_2$, so that $u = s_1[x/s_2]\mathtt{L}$, then the result trivially holds for any possible location of b verifying $a \not\leq_{\mathtt{B}} b$, i.e.: $b \subseteq s_1$, $b \subseteq \mathtt{L}$, $s_1 = C[\![z^{\beta}]\!]$ where $\mathtt{L} = \mathtt{L}_1[z/s_3]\mathtt{L}_2$, and $\mathtt{L} = \mathtt{L}_1[z^{\beta}/s_3]\mathtt{L}_2$.
- If $t = (\lambda x^{\beta}.s_1)\mathtt{L}s_2$, so that $u = (\lambda x^{\beta}.s_1')\mathtt{L}'s_2'$, then we conclude immediately.
- If $t = C[\![x^{\alpha}]\!][x/s_2]$, so that $u = C[\![s_2]\!][x/s_2]$, then $a \not\leq_{\mathtt{B}} b$ implies $b \subseteq C[\![x^{\alpha}]\!]$ or $C = D[x^{\alpha}, x^{\beta}]$. In both cases, we conclude immediately.
- If $t = s_1[x^{\alpha}/s_2]$, then we conclude immediately.
- If $t = C[\![x^{\beta}]\!][x/s_2]$, then we analyse different cases separately:
  - if $\alpha \subseteq s_1$ then: if the occurrence of x^{β} is in the box of the pattern of a then we would contradict $a \not\leq_{\mathtt{B}} b$, otherwise Lem. 4.5.6 allows to conclude;
  - if $\alpha \subseteq s_2$ then we conclude immediately;
  - if $C = D[x^{\alpha}, \Box]$, so that $C[\![x^{\beta}]\!] = D[x^{\alpha}, x^{\beta}]$, then $u = D[s_2, x^{\beta}][x/s_2]$, and this observation suffices to conclude.
- If $t = s_1[x^{\beta}/s_2]$, so that $a \subseteq s_1$ or $a \subseteq s_2$, then the result holds trivially.

The inductive case is when $a \not\bowtie t$ and $b \not\bowtie t$, that is, $t = \lambda x.t_1$, $t = t_1t_2$ or $t = t_1[x/t_2]$, and $a \subseteq t_i$, $b \subseteq t_j$ for some i, j. In this case, if $i = j$ then IH suffices to conclude, and if $i \neq j$ then the result is immediate. Thus we conclude. □

The proof of Context-Freeness involves a case analysis on the possible positions of steps, far more extensive than that just developed for Linearity. Several auxiliary lemmas are needed.

Lemma 4.5.8. *Let $t = (\lambda x^{\alpha}.t_1)[y_1/s_1]\ldots[y_n/s_n]t_2$ and $u = t_1[x/t_2][y_1/s_1]\ldots[y_n/s_n]$, so that $t \xrightarrow{a} u$ where a is the redex labeled by α, and $b, c \in \mathcal{RO}(t)$, $b', c' \in \mathcal{RO}(u)$ such that $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Then $b \prec^1_{\mathtt{B}} c$ iff $b' \prec^1_{\mathtt{B}} c'$.*

Proof. See Appendix C.5, page 273. □

Lemma 4.5.9. *Let $t = E[\![x^\gamma]\!][x/s]$, c the γ-labeled step in t, $a, b \subseteq E[\![x^\gamma]\!]$, $b[\![a]\!]b'$, and $c[\![a]\!]c'$. If $a \not\prec_{\mathtt{B}} c$, then $b \prec^1_{\mathtt{B}} c$ iff $b' \prec^1_{\mathtt{B}} c'$.*

Proof. See Appendix C.5, page 273. □

Lemma 4.5.10. *Let $a, b, c \in \mathcal{RO}(t)$. Assume $a \not\prec_{\mathtt{B}} c$, $t \xrightarrow{a} t'$, $b[\![a]\!]b'$, $c[\![a]\!]c'$ and $b' \prec^n_{\mathtt{B}} d' \prec^1_{\mathtt{B}} c'$, where d' is a created redex. Then $b' \prec^k_{\mathtt{B}} c'$ with $k \leqslant n$.*

Proof. See Appendix C.5, page 275. □

Lemma 4.5.11. *Let u be a term s.t. $x \notin \mathtt{fv}(u)$ and $\gamma \notin \mathtt{Lab}(u)$, and $b \in \mathcal{RO}(u)$. Let E be a context s.t. $c \in \mathcal{RO}(E[\![x]\!])$ has label γ. Then $b \not\prec^1_{\mathtt{B}} c$ in $E[\![u]\!]$.*

Proof. We just conclude by observing that $b \prec^1_{\mathtt{B}} c$ in $E[\![u]\!]$ would imply the label γ occurs in the box of b, therefore in u. □

Lemma 4.5.12. *Let E be a context, u a term, and $b, c \in \mathcal{RO}(E[\![x]\!])$, where b and c are labeled with β and γ respectively. Then $b \prec^1_{\mathtt{B}} c$ iff $b' \prec^1_{\mathtt{B}} c'$, where b' and c' are the β- and γ-labeled steps in $E[\![u]\!]$.*

Proof. Straightforward induction on E. □

Lemma 4.5.13. *Let $a, b \in \mathcal{RO}(t)$, where a and b are labeled with α and β respectively, and E a context. Then $a \prec^1_{\mathtt{B}} b$ iff $a' \prec^1_{\mathtt{B}} b'$, where a' and b' are the α- and β-labeled steps in $E[\![t]\!]$.*

Proof. Straightforward induction on E. □

Proposition 4.5.14 (Context-Freeness for $\mathfrak{A}_{\mathtt{B}}$)**.** *Let a, b, c be coinitial redexes* s.t. *$b[\![a]\!]b'$ and $c[\![a]\!]c'$. If $a \not\prec_{\mathtt{B}} c$ then $(b \prec_{\mathtt{B}} c \Leftrightarrow b' \prec_{\mathtt{B}} c')$.*

Proof. Let us define $t \xrightarrow{a} u$, and consider the variant of t in which a, b and c are given the labels α, β and γ respectively. Notice that $a \in \{b, c\}$ would contradict the existence of b' or c'. Therefore we can assume $a \notin \{b, c\}$ as well as $a \not\prec_{\mathtt{B}} c$.

We prove first that $b \prec^1_{\mathtt{B}} c$ iff $b' \prec^1_{\mathtt{B}} c'$. We proceed by induction on the context of a. The base case is when that context is □. We analyse the different rewrite rules.

Assume $t = (\lambda x^\alpha.t_1)\,\mathtt{L}\,t_2 \xrightarrow{\alpha} t_1\,[x/t_2]\,\mathtt{L} = u$. We conclude by Lemma 4.5.8.

Assume $t = t_1[x^\alpha/t_2] \xrightarrow{\alpha} t_1 = u$. If $b, c \subseteq t_1$, then we conclude immediately by Lemma 4.5.13. Otherwise $b \subseteq t_2$ or $c \subseteq t_2$, contradicting the existence of b' or c'.

Assume that $t = E[\![x^\alpha]\!][x/t_2] \xrightarrow{\alpha} E[\![t_2]\!][x/t_2] = u$. We analyse the possible locations of γ and β. We start by observing that neither b nor c can be $\mathtt{gc}$-steps on the variable x. Moreover, $c \subseteq t_2$ contradicts $a \not\prec_{\mathtt{B}} c$, hence we assume $c \subseteq E[\![x^\alpha]\!]$ or $c \bowtie t$ in the following analysis.

- If $b, c \subseteq E[\![x^\alpha]\!]$, then we conclude by Lemma 4.5.12.

- If $b \subseteq E[\![x^\alpha]\!]$ but $c \bowtie t$, then x^γ occurs free in $E[\![x^\alpha]\!]$.

  Suppose $b \prec^1_{\mathtt{B}} c$. Then the box of b does not contain the whole pattern of c (which is the whole term t), therefore x^γ occurs free in the box of b in t. That is, $E[\![x^\alpha]\!] = D_1[D_2[\![x^\gamma]\!], x^\alpha]$ or $E[\![x^\alpha]\!] = D_3[D_4[x^\gamma, x^\alpha]]$, where the pattern of b is $D_2[\![x^\gamma]\!]$ or $D_4[x^\gamma, x^\alpha]$. Hence $u = D_1[D_2[\![x^\gamma]\!], t_2][x/t_2]$ or $u = D_3[D_4[x^\gamma, t_2]][x/t_2]$. It is straightforward to observe that x^γ also lies inside the box of b in u, therefore $b' \prec^1_{\mathtt{B}} c'$.

  Suppose $b' \prec^1_{\mathtt{B}} c'$. Then a similar analysis applies, observing that $x \notin \mathtt{fv}(t_2)$, to conclude $b \prec^1_{\mathtt{B}} c$.

- If $b \bowtie t$ (x^β occurs free in $E[\![x^\alpha]\!]$) and either $c \subseteq E[\![x^\alpha]\!]$ or $c \bowtie t$, then $b \not\prec^1_{\mathtt{B}} c$ and $b' \not\prec^1_{\mathtt{B}} c'$ since the label γ does not occur in the box of the β-labeled step (which is t_2) in neither t nor u.

- If $b \subseteq t_2$ and either $c \subseteq E[\![x^\alpha]\!]$ or $c \bowtie t$, then $t_2 = D[\![p]\!]$ and $u = E[\![D[\![p]\!]]\!][x/D[\![p]\!]]$ where p is the pattern of b. Let us call b_0 (resp. b_1) the step whose pattern is the leftmost (resp. rightmost) occurrence of p in u. Then $b[\![a]\!]b_0$ and $b[\![a]\!]b_1$.

  We first remark that $b \not\prec^1_{\mathtt{B}} c$ and $b_1 \not\prec^1_{\mathtt{B}} c'$ since the label γ neither occurs in the box of b in t nor in that of b_1 in u. Moreover, if $c \subseteq E[\![x^\alpha]\!]$, then $b_0 \not\prec^1_{\mathtt{B}} c'$ holds by Lemma 4.5.11 observing that the γ label does not occur in t_2; if $c \bowtie t$, i.e. x^γ occurs free in $E[\![x^\alpha]\!]$, then $b_0 \not\prec^1_{\mathtt{B}} c'$ because the free occurrence of x^γ in $E[\![t_2]\!]$ is not inside t_2. Thus we conclude.

Now we analyse the inductive cases.

- $t = \lambda x.t_1 \xrightarrow{\alpha} \lambda x.u_1 = u$, where $t_1 \xrightarrow{\alpha} u_1$. We conclude by the IH on t_1 and Lemma 4.5.13.

- $t = t_1 t_2 \xrightarrow{\alpha} u_1 t_2 = u$ or $t = t_1 t_2 \xrightarrow{\alpha} t_1 u_2 = u$, where $t_i \xrightarrow{\alpha} u_i$ for $i \in \{1, 2\}$.

  If b and c lie in different t_js, i.e. $b \subseteq t_j$ and $c \subseteq t_{3-j}$ for $j \in \{1, 2\}$, then $b \not\prec^1_{\mathtt{B}} c$ and $b' \not\prec^1_{\mathtt{B}} c'$ since the box of b (resp. b') does not contain the label γ.

  If $b, c \subseteq t_i$, then we conclude by the IH on t_i.

  If $b, c \subseteq t_{3-i}$, then we conclude by Lemma 4.5.13.

  If $b \bowtie t$, i.e. $t_1 = (\lambda x^\beta.t_{11})\,\mathtt{L}$, then $b \prec^1_{\mathtt{B}} c$ iff $b' \prec^1_{\mathtt{B}} c'$ iff $c \subseteq t_2$.

  If $c \bowtie t$, i.e. $t_1 = (\lambda x^\gamma.t_{11})\,\mathtt{L}$, then $b \not\prec^1_{\mathtt{B}} c$ and $b' \not\prec^1_{\mathtt{B}} c'$.

- $t = t_1[x/t_2] \xrightarrow{\alpha} u_1[x/t_2] = u$ or $t = t_1[x/t_2] \xrightarrow{\alpha} t_1[x/u_2] = u$, where $t_i \xrightarrow{\alpha} u_i$ for $i \in \{1, 2\}$.
  - If $b, c \subseteq t_i$, then we conclude by the IH on t_i.
  - If $b, c \subseteq t_{3-i}$, then we conclude by Lemma 4.5.13.
  - If $c \subseteq t_j$ and $b \subseteq t_{3-j}$ for $j = 1, 2$, then $b \not\prec^1_{\mathtt{B}} c$ and $b' \not\prec^1_{\mathtt{B}} c'$ since the box of b (resp. b') does not contain the label γ.
  - If $b \bowtie t$ and $c \subseteq t_2$, i.e. either x^β occurs free in t_1 or $[x/t_2]$ is indeed $[x^\beta/t_2]$, then $b \prec^1_{\mathtt{B}} c$ and $b' \prec^1_{\mathtt{B}} c'$. If $i = 1$ and x^β occurs in t_1, then observe that the existence of b' implies that x^β occurs in u_1.
  - If $b \bowtie t$ and $c \subseteq t_1$, then $b \not\prec^1_{\mathtt{B}} c$ and $b' \not\prec^1_{\mathtt{B}} c'$ since the box of b (resp. b') does not contain the label γ.

- If $c \bowtie t$ is the gc-step on x, so that $x \notin \mathtt{fv}(t_1)$, implying eventually $x \notin \mathtt{fv}(u_1)$, then the box of no step in t or u contains the label γ, so that $b \not\prec^1_{\mathtt{B}} c$ and $b' \not\prec^1_{\mathtt{B}} c'$.
- If $c \bowtie t$ is a ls-step, i.e. x^γ occurs free in t_1, there are several cases:

  If $b \subseteq t_1$ and $i = 2$, then it suffices to observe that t_1 remains unchanged.

  If $b \subseteq t_1$ and $i = 1$ so that $u_1 = D[\![x^\gamma]\!]$ (notice that Linearity and $a \not\prec_{\mathtt{B}} c$ imply x^γ occurs free exactly once in u_1) then we conclude by Lemma 4.5.9 by using the condition $a \not\prec_{\mathtt{B}} c$ which is needed in order to apply that lemma.

  Otherwise, $b \not\prec^1_{\mathtt{B}} c$ and $b' \not\prec^1_{\mathtt{B}} c'$ since the box of b (resp. b') does not contain the label γ.

We now prove the statement of the lemma, i.e. that $b \prec_{\mathtt{B}} c$ iff $b' \prec_{\mathtt{B}} c'$. We prove each side of the iff separately, proceeding by induction on n in $b \prec^n_{\mathtt{B}} c$, resp. $b' \prec^n_{\mathtt{B}} c'$.

Assume $b \prec^{n-1}_{\mathtt{B}} d \prec^1_{\mathtt{B}} c$. Observing $a \prec_{\mathtt{B}} d$ would imply $a \prec_{\mathtt{B}} c$, we obtain $a \not\prec_{\mathtt{B}} d$. Linearity then gives the existence of exactly one d' s.t. $d[\![a]\!]d'$. We apply IH on $n-1$, thus obtaining $b' \prec_{\mathtt{B}} d'$. Moreover, $d' \prec^1_{\mathtt{B}} c'$ holds by the proof we have just performed for the $\prec^1_{\mathtt{B}}$ case. Thus we conclude.

Assume $b' \prec^{n-1}_{\mathtt{B}} d' \prec^1_{\mathtt{B}} c'$, where $d' \in \mathcal{RO}(u)$ and $d[\![a]\!]d'$ for some $d \in \mathcal{RO}(t)$. By the proof we have just performed for the $\prec^1_{\mathtt{B}}$ case we obtain $d \prec^1_{\mathtt{B}} c$, so that again we get $a \not\prec_{\mathtt{B}} d$. Then we proceed similarly to the previous case.

Assume $b' \prec^{n-1}_{\mathtt{B}} d' \prec^1_{\mathtt{B}} c'$, where $d' \in \mathcal{RO}(u)$ is a created redex. Lemma 4.5.10 implies $b' \prec^k_{\mathtt{B}} c'$ for some $k \leqslant n-1$, so that we conclude by the IH on k. □

As we have already remarked, to analyse the steps, residuals and box embedding for an object of $\mathfrak{A}^{\sim}_{\mathtt{B}}$, namely a $\sim$-equivalence class, it is enough to observe an arbitrary term belonging to that class. Therefore, the proofs of the initial axioms, FD and SO given for $\mathfrak{A}_{\mathtt{L}}$, and those of Linearity and Context-Freeness, given for $\mathfrak{A}_{\mathtt{B}}$, apply to $\mathfrak{A}^{\sim}_{\mathtt{B}}$ as well. For FD, we notice that the measures used in the proof, namely the functions $\mathtt{LM}_x$ and PLR, are stable by $\sim$; cfr. Lem. 4.2.13. Therefore, these measures can also be considered as defined on $\sim$-equivalence classes.

4.5.2 Some standardisation results stable by graphical equivalence

As described in Section 2.1.8, holding the initial, FD, SO, Linearity and Context-Freeness axioms, are a sufficient condition, for an ARS, to assert the existence of a s.r.s. being permutation equivalent to a reduction sequence. Therefore, the latter result holds for $\mathfrak{A}_{\mathtt{B}}$ and $\mathfrak{A}^{\sim}_{\mathtt{B}}$. Namely:

Theorem 4.5.15. *Let γ be a reduction sequence in $\mathfrak{A}_{\mathtt{B}}$. Then there exists a* s.r.s. *δ, such that δ and γ are permutation equivalent.*

Theorem 4.5.16. *Let γ be a reduction sequence in $\mathfrak{A}^{\sim}_{\mathtt{B}}$. Then there exists a* s.r.s. *δ, such that δ and γ are permutation equivalent.*

Proof. Both theorems are immediate corollaries of Thm. 2.1.23, given the axiom proofs presented in Section 4.2 and Section 4.5.1. □

On the other hand, the *uniqueness* of `s.r.s.` modulo square equivalence cannot be proved for $\mathfrak{A}_{\mathtt{B}}$, and hence neither for $\mathfrak{A}_{\widetilde{\mathtt{B}}}$, by resorting to Thm. 2.1.24, the uniqueness result given in Section 2.1.8. The reason is that these ARS, contrarily to $\mathfrak{A}_{\mathtt{L}}$ (cfr. Thm. 4.3.1), do not enjoy the required axioms. A case analysis entailing a counterexample for Enclave–Creation, and also one for Enclave–Embedding, follows.

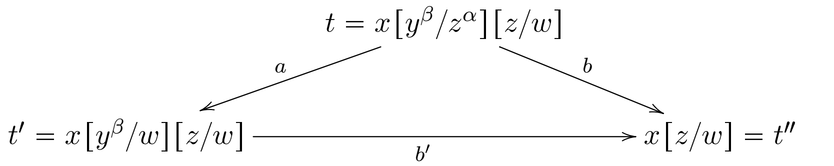
Consider $t = ((\lambda x^{\beta}.x)(y[z^{\alpha}/z']))[z'/u] \xrightarrow{a} ((\lambda x^{\beta}.x)y)[z'/u] = t'$, where a and b are the steps in t whose anchors are labeled with α and β respectively, and d is a step in t verifying $d \subseteq u$. Let us define c' as the `gc`-step for z', created by a, and b', d' the unique steps in t' verifying $b[\![a]\!]b'$ and $d[\![a]\!]d'$. Then we have $b \prec_{\mathtt{B}} a$ and $b' \not\prec_{\mathtt{B}} c'$, and also $b \prec_{\mathtt{B}} a \prec_{\mathtt{B}} d$ (more precisely, $a \prec^2_{\mathtt{B}} d$, since $a \prec^1_{\mathtt{B}} e \prec^1_{\mathtt{B}} d$ where e is the `ls`-step on the occurrence of z' in $[z^{\alpha}/z']$) and $b' \not\prec_{\mathtt{B}} d'$. Thus both Enclave–Creation and Enclave–Embedding are contradicted.

The preceding counterexample involves the `gc`-rule. We notice that the calculus generated only by the `db`- and `ls`-rules does not enjoy Enclave–Embedding either. Consider

$$t = ((\lambda x^{\beta}.x)y^{\alpha})[y/w^{\gamma}[w/z]] \quad \xrightarrow{a} \quad ((\lambda x^{\beta}.x)w^{\gamma}[w/z])[y/w^{\gamma}[w/z]] = t'$$

and let a, b, c be the steps in t labeled with α, β, γ respectively; notice $b \prec_{\mathtt{B}} a \prec_{\mathtt{B}} c$. Then we have $b[\![a]\!]b'$, $c[\![a]\!]c''$ and $b' \not\prec_{\mathtt{B}} c''$, where b' is the β-labeled step in t', and c'' the rightmost γ-labeled one.

W.r.t. Stability, we observe that there is a case in which a step can be created in two different ways, but not being a counterexample for the axiom, since the two steps involved are not disjoint. Consider



where again, the anchors of a and b are labeled with α and β respectively. In this case, both a and b create the `gc`-step on z. On the other hand, $b \prec_{\mathtt{B}} a$, so that this case is not a counterexample for Stability. Notice that there are no ambiguity about the standard way to go from t' to t'', in line with the discussion in page 121 preceding the definition of the box order: b is a `s.r.s.`, while $a; b'$ is not. We conjecture that the Stability axiom is valid for $\mathfrak{A}_{\mathtt{B}}$ and $\mathfrak{A}_{\widetilde{\mathtt{B}}}$, we did not try to build a proof yet.

In the next section, we introduce a novel technique allowing to obtain standardisation uniqueness results for $\mathfrak{A}_{\mathtt{B}}$ and $\mathfrak{A}_{\widetilde{\mathtt{B}}}$.

4.6 A novel proof for the uniqueness of `s.r.s.`

In this section, we develop an abstract proof of standardisation, which allows to obtain the result of uniqueness of `s.r.s.`, modulo square equivalence, for an ARS $\mathfrak{A}$. This proof does not require $\mathfrak{A}$ to verify any of the Enclave–Creation, Enclave–Embedding or Stability axioms. The initial axioms, FD, SO, Linearity and Context-Freeness are the requirements imposed on $\mathfrak{A}$.

The proof relies in the existence of a second ARS, let us call it $\mathfrak{A}'$, whose terms, steps and residuals coincides with that of $\mathfrak{A}$, whose embedding relation is a total order containing that of $\mathfrak{A}$, and which verifies all the standardisation axioms. Therefore, the uniqueness of `s.r.s.` for $\mathfrak{A}'$ can be obtained as a corollary of Thm. 2.1.24. On the other hand, the *existence* of a `s.r.s.` equivalent to any reduction sequence in $\mathfrak{A}$ yields from Thm. 2.1.23. These facts are exploited to obtain the uniqueness of `s.r.s.` result for $\mathfrak{A}$.

This proof applies to $\mathfrak{A}_{\mathtt{B}}$, where the ARS $\mathfrak{A}_{\mathtt{L}}$, whose embedding is the total order $\prec_{\mathtt{L}}$, plays the role of $\mathfrak{A}'$ in the abstract proof.

The following notion is referred to in the forthcoming statements and proofs: given $t \xrightarrow{\delta}\!\!\!\!\rightarrow t'$ and $a \in \mathcal{RO}(t)$, we say that a **is contracted along** δ iff $\delta = \delta_1; a'; \delta_2$ where $a[\![\delta_1]\!]a'$.

Two auxiliary results are needed, namely:

Lemma 4.6.1. *Let $\mathfrak{A}$ be an ARS satisfying the initial axioms, FD, SO, and all the embedding axioms. We note its embedding relation as $<$. Let $t \xrightarrow{\delta}\!\!\!\!\rightarrow t' \xrightarrow{c'} u$, and $a \in \mathcal{RO}(t)$ such that $a < b$ for all $b \in \mathcal{RO}(t)$ contracted along $\delta; c'$. Then there exists $a' \in \mathcal{RO}(t')$ such that $a[\![\delta]\!] = \{a'\}$ and $a' < c'$.*

Proof. We proceed by induction on $|\delta|$.

If $|\delta| = 0$, i.e. $\delta = \mathtt{nil}_t$, then $t' = t$, $t \xrightarrow{c'} u$, and $c' \in \mathcal{RO}(t)$ is obviously contracted along itself, so that the hypothesis implies $a < c'$. We conclude by observing that $a[\![\delta]\!] = \{a\}$.

If $\delta = d; \delta'$, then $t \xrightarrow{d} t_0 \xrightarrow{\delta'}\!\!\!\!\rightarrow t' \xrightarrow{c'} u$. The hypothesis implies $a < d$, then Linearity yields $a[\![d]\!] = \{a_0\}$. Let $b_0 \in \mathcal{RO}(t_0)$ contracted along $\delta'; c'$.

- If $b[\![d]\!]b_0$ for some $b \in \mathcal{RO}(t)$, then b is contracted along $\delta; c'$, so that the hypothesis implies $a < b$.

  If $d \not< b$, then Context-Freeness yields $a_0 < b_0$.

  If $d < b$, then $a < d < b$; in this case we obtain $a_0 < b_0$ by Enclave–Embedding.
- If $\varnothing[\![d]\!]b_0$, then Enclave–Creation implies $a_0 < b_0$.

Hence the IH applies to $|\delta'|$, which suffices to conclude. □

Lemma 4.6.2. *Let $\mathfrak{A}$ be an ARS verifying the initial axioms, FD, SO, and all the embedding axioms, whose embedding relation, noted $<$, is a total order. Let $t \xrightarrow{a} t_a \xrightarrow{\delta}\!\!\!\!\rightarrow t'$, where δ is a `s.r.s.` for $\mathfrak{A}$, and $a < b$ for any $b \in \mathcal{RO}(t)$ contracted along $a; \delta$ and verifying $b \neq a$. Then $a; \delta$ is a `s.r.s.` for $\mathfrak{A}$.*

Proof. If $\delta = \mathtt{nil}_t$ then we conclude immediately, therefore we assume $\delta = b_0; \delta'$.

As the square equivalence of $\mathfrak{A}$ is the identity, then $a; \delta$ not being $\mathfrak{A}$-standard would imply the existence of an anti-standard pair in that reduction sequence. On the other hand, standardness of δ implies that it does not include any anti-standard pair. Consequently, verifying that $a; b_0$ is not an anti-standard pair suffices to conclude.

If $b[\![a]\!]b_0$, then the hypothesis implies $a < b$. Otherwise $\varnothing[\![a]\!]b_0$. In both cases, $a; b_0$ is not anti-standard. Thus we conclude. □

Theorem 4.6.3. *Let $\mathfrak{A}_1$, $\mathfrak{A}_2$ be two ARS verifying the following conditions, where for $i = 1, 2$, $<_i$ is the embedding relation of $\mathfrak{A}_i$:*

- *the sets of objects and steps, the source and target functions, and the residual relation, coincide for $\mathfrak{A}_1$ and $\mathfrak{A}_2$.*

- *The embedding relation* $<_1$ *is a total order.*
- $<_2 \subseteq <_1$.
- $\mathfrak{A}_1$ *and* $\mathfrak{A}_2$ *verify the initial axioms, FD, SO, Linearity and Context-Freeness.*
- $\mathfrak{A}_1$ *verifies the rest of the embedding axioms as well.*

Then the uniqueness of `s.r.s.` *result stated in Thm. 2.1.24 holds for* $\mathfrak{A}_2$*. Namely, for any reduction sequence* γ*, there is a* `s.r.s.` δ *such that* δ *and* γ *are permutation equivalent, and moreover* δ *is unique modulo square equivalence. That is, for any* `s.r.s.` δ'*,* δ' *being permutation equivalent with* γ *implies* $\delta \diamond \delta'$*.*

Proof. Observe that any reduction sequence for $\mathfrak{A}_1$ is a reduction sequence for $\mathfrak{A}_2$ and vice versa. Moreover, permutation equivalence also coincide for both ARSs, since the definition of permutation equivalence for ARS, cfr. Section 2.1.7, does not depend on the embedding relation.

Let γ be a reduction sequence for both ARSs. Let δ_1 and δ_2 be two reduction sequences, such that both are permutation equivalent with γ, δ_1 is a `s.r.s.` for $\mathfrak{A}_1$ and δ_2 is a `s.r.s.` for $\mathfrak{A}_2$. The existence of δ_1 and δ_2 are a consequence of Thm. 2.1.23. Moreover, δ_1 is unique modulo $\diamond$, by Thm. 2.1.24. Furthermore, $<_1$ being a total order implies that the square equivalence for $\mathfrak{A}_1$ is the identity; cfr. the comment in the proof of Thm. 4.3.1. Hence, δ_1 is *unique* given that it is a `s.r.s.` for $\mathfrak{A}_1$ and is permutation equivalent with γ.

To conclude, it suffices to show $\delta_1 \diamond \delta_2$: any reduction sequence δ_2' being permutation equivalent with γ and `s.r.s.` for $\mathfrak{A}_2$, would verify $\delta_1 \diamond \delta_2'$ as well, and therefore $\delta_2 \diamond \delta_2'$ by transitivity of $\diamond$.

We prove $\delta_1 \diamond \delta_2$ by induction on $|\delta_2|$. Let us define t' as the term verifying $t \xrightarrow{\gamma}\!\!\!\!\rightarrow t'$, $t \xrightarrow{\delta_1}\!\!\!\!\rightarrow t'$ and $t \xrightarrow{\delta_2}\!\!\!\!\rightarrow t'$.

If $|\delta_2| = 0$, i.e. $\delta_2 = \mathtt{nil}_t$, then $\delta_1 = \gamma = \delta_2$ since all of them are permutation equivalent. Thus we conclude.

If $\delta_2 \neq \mathtt{nil}_t$, then we consider the minimal, w.r.t. $<_1$, of the steps in $\mathcal{RO}(t)$ contracted along δ_2; let us call this step a. That is, a is contracted along δ_2, and for any $b \in \mathcal{RO}(t)$ contracted along δ_2 such that $b \neq a$, $a <_1 b$.

Let $\delta_2 = \delta'; a'; \delta''$ where $a[\![\delta']\!]a'$, so that $t \xrightarrow{\delta'}\!\!\!\!\rightarrow t_1 \xrightarrow{a'} t_a \xrightarrow{\delta''}\!\!\!\!\rightarrow t'$. We verify that $\delta_2 \diamond a; \delta'[\![a]\!]; \delta''$ and $|\delta'[\![a]\!]| = |\delta'|$, by induction on $|\delta'|$.

- If $\delta' = \mathtt{nil}_t$, so that $t_1 = t$ and $a \in \mathcal{RO}(t)$, then $\delta'[\![a]\!] = \mathtt{nil}_{t_a}$, and $a[\![\delta']\!]a$. In this case, $\delta_2 = a; \delta'' = a; \delta'[\![a]\!]; \delta''$, hence we conclude.
- If $\delta' = \pi; b$, then let a_1 be the step verifying $a[\![\pi]\!]a_1[\![b]\!]a'$. In this case, $\delta_2 = \pi; b; a'; \delta''$. Observe that $a <_1 c$ for any $c \in \mathcal{RO}(t)$ contracted along $\pi; b$. Then Lem. 4.6.1 applies, yielding $a[\![\pi]\!] = \{a_1\}$ and $a_1 <_1 b$, and therefore $b \not<_2 a_1$. On the other hand, $a_1 <_2 b$ would contradict the $\mathfrak{A}_2$-standardness of δ_2. Consequently, $a_1 \parallel_2 b$, so that Linearity yields $a_1[\![b]\!] = \{a'\}$ and $b[\![a_1]\!] = \{b'\}$ for some step b'. Therefore, $\delta_2 \diamond \pi; a_1; b'; \delta''$, implying that the latter is $\mathfrak{A}_2$-standard.

  Observe that the sets of steps in t contracted along the two equivalent reduction sequences δ_2 and $\pi; a_1; b'; \delta''$ coincide, so that a is the minimal, w.r.t. $<_1$, step in such a set for $\pi; a_1; b'; \delta''$. This observation implies that IH applies on π, yielding:

$$\delta_2 \;\diamond\; \pi; a_1; b'; \delta'' \;\diamond\; a; \pi[\![a]\!]; b'; \delta''$$

and $|\pi[\![a]\!]| = |\pi|$. We conclude by observing that $\delta'[\![a]\!] = \pi[\![a]\!]; b'$.

We have thus obtained $\delta_2 = \delta'; a; \delta'' \diamond a; \delta'[\![a]\!]; \delta''$, and $|\delta_2| = |a; \delta'[\![a]\!]; \delta''|$. Then $a; \delta'[\![a]\!]; \delta''$ is standard for $\mathfrak{A}_2$. It is straightforward to obtain that $\delta'[\![a]\!]; \delta''$ is standard for $\mathfrak{A}_2$ as well.

Let θ be the unique $\mathfrak{A}_1$-`s.r.s.` permutation equivalent to $\delta'[\![a]\!]; \delta''$. Then the IH on $|\delta_2|$ applies, because $|\delta'[\![a]\!]; \delta''| < |\delta_2|$. We obtain $\theta \diamond \delta'[\![a]\!]; \delta''$, and therefore $a; \theta \diamond a; \delta'[\![a]\!]; \delta'' \diamond \delta_2$. As we already noticed, the set of steps in $\mathcal{RO}(t)$ for δ_2 and $a; \theta$ coincide. Hence a is the $<_1$-minimal such step for $a; \theta$, so that Lem. 4.6.2 applies, yielding that $a; \theta$ is $\mathfrak{A}_1$-standard. Moreover, it is permutation equivalent with γ, and therefore with δ_1. Hence, uniqueness of `s.r.s.` for $\mathfrak{A}_1$ yields $\delta_1 = a; \theta \diamond \delta_2$. Thus we conclude. □

This proof allows to obtain strong standardisation results for $\mathfrak{A}_{\mathtt{B}}$ and, consequently, for $\mathfrak{A}_{\mathtt{B}}^{\sim}$.

Theorem 4.6.4. *Let γ be a reduction sequence in $\mathfrak{A}_{\mathtt{B}}$. Then there exists a `s.r.s.` δ such that γ and δ are permutation equivalent. Moreover, δ is* unique *modulo $\diamond$. That is, for any `s.r.s.`δ', δ' being permutation equivalent with γ implies $\delta \diamond \delta'$.*

Proof. This statement is a corollary of Thm. 4.6.3, where $\mathfrak{A}_{\mathtt{L}}$ and $\mathfrak{A}_{\mathtt{B}}$ play the roles of $\mathfrak{A}_1$ and $\mathfrak{A}_2$ respectively. The required axioms are verified in Section 4.2 and Section 4.5.1 respectively. □

Theorem 4.6.5. *Let γ be a reduction sequence in $\mathfrak{A}_{\mathtt{B}}^{\sim}$. Then there exists a `s.r.s.` δ such that γ and δ are permutation equivalent. Moreover, δ is* unique *modulo $\diamond$. That is, for any `s.r.s.`δ', δ' being permutation equivalent with γ implies $\delta \diamond \delta'$.*

Proof. Thm. 4.5.16 yields the existence of δ. Since reduction sequences, residuals and embedding for a $\sim$-equivalence class can be observed on any term belonging to that class, then two permutation equivalent `s.r.s.` δ and δ' can be observed on the same term, so that Thm. 4.6.4 implies $\delta \diamond \delta'$. Thus we conclude. □

Chapter 5

Permutation equivalence for infinitary rewriting

As described in Section 2.2, proof terms are a representation of reduction sequences, and more generally of different forms of *contraction activity.* Each rewriting rule is denoted by an ad-hoc symbol, while the binary symbol $\cdot$ denotes concatenation. Equivalence of contraction activities is characterised by equational logic based on proof terms: two contraction activities are said *permutation equivalent* iff the proof terms representing them can be proven equal by the contextual and equivalence closure of a set of basic equations.

In this chapter, we generalise the notion of proof term to the realm of *infinite reduction*; more precisely, to first-order, left-linear infinitary term rewriting systems, adopting the *strong convergence* criterion; cfr. Section 1.2.3. We define and study the notion of permutation equivalence for transfinite reductions, by means of the so defined *infinitary proof terms.*

Infinity arises in different ways in infinitary rewriting. Consider the rewriting rule $f(x) \to g(x)$, and the reduction sequence

$$f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega$$

where f^ω is a concise description of the term $f(f(f(\ldots)))$, which can be also described by the equation $t = f(t)$. This reduction sequence comprises the application of the given rewriting rule to each of the occurrences of f in the source term f^ω. This term includes an *infinite* number of such occurrences, implying that the length of the given reduction sequence is infinite. On the other hand, observe that all the steps included in this sequence correspond to redexes present in the source term (put in other words, no *created* redex is contracted); therefore, this reduction sequence is equivalent to the simultaneous contraction of an *infinite* set of steps:

$$f^\omega \multimap g^\omega$$

where $\multimap$ denotes the simultaneous contraction of a number of steps, in this case an infinite number of them. The transformation of each of the f in f^ω into a g can be organised in many other different ways, e.g.:

$$\begin{aligned}
&f^\omega \multimap g(g(f^\omega)) \multimap g(g(g(g(f^\omega)))) \multimap\!\!\!\!\twoheadrightarrow g^\omega \\
&f^\omega \to f(g(f^\omega)) \to f(g(f(g(f^\omega)))) \twoheadrightarrow f(g(f(g(f(g(\ldots)))))) \\
&\qquad\qquad\qquad\qquad \to g(g(f(g(f(g(\ldots)))))) \to g(g(g(g(f(g(\ldots)))))) \twoheadrightarrow g^\omega
\end{aligned}$$

For each possible way to transform f^ω to g^ω through the given rule, an infinitary proof term denoting precisely that contraction activity must exist. In turn, we should be able to conclude that all the resulting proof terms are permutation equivalent.

This example shows that the extension of the notion of proof term to infinitary rewriting should consider infinite concatenations of steps, or more generally of contraction activities, including the possibility of going beyond the ω-th component, as in the last given reduction sequence, in which a concatenation of ω steps is followed by another one of the same length. The simultaneous contraction of an infinite number of steps must be taken into account as well.

In the *permutation equivalence* perspective of the equivalence between reductions, two reduction sequences are considered equivalent iff each of them can be transformed into the other by means of a number of permutations of adjacent steps. The characterisation of equivalence in both the ARS and the proof term models we use in this thesis reflect this view, cfr. Sections 2.1.7 and Section 2.2.[1]

Infinitary rewriting leads to the existence of infinite, equivalent reduction sequences, where the transformation of one of them into the other one involves an infinite number of step permutations. Recalling an example from Section 1.4.3, consider the rules $f(x) \to g(x)$ and $m(x) \to n(x)$. In order to verify the equivalence between the following two reductions:

$$\begin{array}{l} m(f^\omega) \to m(g(f^\omega)) \to m(g(g(f^\omega))) \twoheadrightarrow m(g^\omega) \to n(g^\omega) \\ m(f^\omega) \to n(f^\omega) \to n(g(f^\omega)) \to n(g(g(f^\omega))) \twoheadrightarrow n(g^\omega) \end{array}$$

the last step in the former reduction must be permuted with an infinite number of steps, since it corresponds with the first step in the latter reduction.

For a more complex example, we add the rule $g(x) \to j(x)$. Assume that we want to prove the equivalence between the following reduction sequences:

$$f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega \to j(g^\omega) \to j(j(g^\omega)) \twoheadrightarrow j^\omega \tag{5.1}$$

$$f^\omega \to g(f^\omega) \to j(f^\omega) \to j(g(f^\omega)) \to j(j(f^\omega)) \twoheadrightarrow j^\omega \tag{5.2}$$

If we successively permute the step $g^\omega \to j(g^\omega)$ in (5.1) with each of the infinite preceding steps except for the first one, we would obtain

$$f^\omega \to g(f^\omega) \to j(f^\omega) \to j(g(f^\omega)) \to j(g(g(f^\omega))) \twoheadrightarrow j(g^\omega) \to j(j(g^\omega)) \twoheadrightarrow j^\omega$$

To transform this reduction sequence into (5.2), we would need to repeat this process for *each of the infinite* steps corresponding to the rule $g(x) \to j(x)$.

The definitions we give in this chapter take into account the aforementioned considerations. We remark that the equivalence of reduction sequences having *different lengths* can be stated by means of these definitions; e.g., the lengths of the equivalent sequences (5.1) and (5.2) are $\omega \times 2$ and ω respectively. The phenomenon of *infinitary erasure* is adequately reflected as well, cfr. Section 5.3.4.

A finite reduction sequence $\delta = a_1; a_2; \ldots; a_n$ can be represented by a proof term having the form $\psi = \psi_1 \cdot \psi_2 \cdot \ldots \cdot \psi_n$, where each ψ_i describes precisely the step a_i.

[1] We recall that [BKdV03] includes other characterisations of equivalence as well, related with notions different from permutation of steps.

Each ψ_i includes exactly one rule symbol, and does not contain occurrences of the dot. Such representation of a given rewrite step a_i is unique. We give an example in Fig. 5.1.

$$\text{Rules: } \mu : f(x) \to g(x) \qquad \tau : h(g(x), y) \to k(y, x) \qquad \pi : a \to b$$

$$\begin{array}{ccccccc} h(f(c), a) & \xrightarrow{a_1} & h(g(c), a) & \xrightarrow{a_2} & h(g(c), b) & \xrightarrow{a_3} & k(b, c) \\ & (\ h(\mu(c), a) & \cdot & h(g(c), \pi)\ ) & \cdot & \tau(c, b) & \\ & (\ \psi_1 & \cdot & \psi_2\ ) & \cdot & \psi_3 & \end{array}$$

Figure 5.1: A reduction sequence, and a proof term representing it

On the other hand, the fact that the dot is a binary symbol implies that its occurrences can be associated in different ways. E.g., observe that the proof term $h(\mu(c), a) \cdot (\, h(g(c), \pi) \cdot \tau(c, b)\,)$ also denotes the reduction sequence given in Fig. 5.1. We say that the representation, in the described way, of a finite reduction sequence as a proof term, is *unique up to rebracketing*, i.e., to the associativity of the dot. Observe that the "equality-up-to-rebracketing" relation on finite proof terms corresponds exactly to the equivalence relation generated solely by the (Assoc) equation schema.

In this chapter, we extend this correspondence between reduction sequences and proof terms to infinitary rewriting, and also formalise it. Namely, we formalise the notion of *infinitary stepwise proof terms*, to wit, the proof terms which precisely denote reduction sequences. The idea of two stepwise proof terms being "equal-up-to-rebracketing" is formalised by the introduction of the *rebracketing equivalence* relation, which is (analogously to the case for finitary rewriting) the infinitary equivalence relation on stepwise proof terms generated by the (Assoc) equation schema. We also formalise the idea of denotation of a reduction sequence by means of a stepwise proof term. Subsequently, we prove that any infinitary reduction sequence whose length is a countable ordinal is denoted by a proof term, and that moreover this proof term is unique up to rebracketing. This result applies particularly to all strong convergent reduction sequences, cfr. [KdV05].

Besides the ability to faithfully represent any strongly convergent reduction sequence, and to reason about the equivalence of reductions, we claim that the proof term model provides a framework in which relevant properties of infinitary rewriting can be proved. We give in this chapter a proof, based on proof terms, of the *compression* result for infinitary rewriting, described in Section 1.2.3. Specifically, we prove that any infinitary proof term is permutation equivalent to a stepwise proof term whose length is at most ω. In this way, we provide a version of the compression result strengthened in two different ways: it comprises not only reduction sequences but also other forms of contraction activity, and it establishes that the original and the compressed reduction are *equivalent*, besides coinciding in their source and target terms. Note that a compression statement establishing the equivalence between the original and compressed reductions is proved in [KKSdV95, BKdV03]; however, orthogonality of the rewriting system is assumed in these results.

The main definitions we present in this chapter are given by *inductive* means, so that inductive arguments can be used to reason about them. We notice the existence of sound induction principles for fundamental concepts in infinitary rewriting. While the number of *occurrences* in an infinite term are indeed infinite, the distance of any of them to the root of the term is finite, giving an induction principle to reason about the set of occurrences of a term. On a different front, while an infinitary reduction sequence can include an infinite *number of steps*, an (either finite or transfinite) ordinal can be set as the length of any sequence, so that transfinite induction can be used to reason on reduction sequences in infinitary rewriting. Strong convergence provides an added element we can use for the limit case in such reasonings: the sequence of the depths of the successive steps, up to any limit ordinal, tends to infinity in a strongly convergent reduction sequence. By "depth of a step" we mean, in this chapter, the depth of the corresponding redex, that is, the distance of the redex to the root of the source term of the step.

We give an adequate, transfinite induction principle to reason on the set of infinitary proof terms, by associating a countable ordinal to each one. This ordinal is related to the occurrences of the concatenation symbol, i.e. the dot: the base case corresponds to the proof terms in which the dot does not occur, and the limit case to the representation of infinite concatenations. An analogous technique is used to reason by transfinite induction on the set of permutation equivalence judgements: a countable ordinal is associated to each judgement. The base case corresponds to the instances of the basic equation schemas. The ordinal associated to the conclusion of a rule is always strictly greater than that of any of its premises.

An alternative, *co-inductive* approach to the study of infinitary rewriting, was proposed in [EHH$^+$13]. Proof objects emerge there as witnesses in the co-inductive characterisation of the reduction relation. Their focus, however, is on techniques for proving properties of the reduction *relation*, rather than the fine structure of the space of transfinite reductions, which is our primary interest. Another coinductive study of infinitary rewriting, also focused in the reduction relation, was recently presented in [Cza14].

We end this introduction with a remark about terminology: in this chapter, we use the acronym "TRS" with the meaning of "first-order term rewriting system"; "iTRS" denotes an infinitary, first-order term rewriting system.

Plan of the chapter

Section 5.1 includes the *preliminary material* needed for the development of the rest of the chapter. Particularly, we give the definitions for the main concepts of infinitary rewriting, including those of term, rewrite step and reduction sequence, we use later on. Section 5.2 is devoted to the notion of *proof term*. We define the set of valid infinitary proof terms. As this definition is involved, we describe extensively its organisation, and we also provide several examples. In Section 5.3, we present a characterisation of the *equivalence of infinitary reductions* based on proof terms, which formalises the notion of permutation equivalence as already noted. We describe, by means of several examples, the challenges imposed by infinite reductions to the formalisation of permutation equivalence; subsequently, we show how the proposed characterisation deals adequately with those examples. In this section we also introduce the phenomenon of *infinitary erasure*, and show that it is accurately modeled by the proposed permutation equivalence characterisation. In Section 5.4, we address the issue of the *denotation of reduction sequences* by means of proof terms, along the lines described earlier in this introduction. Finally, Section 5.5 is devoted to the proof of the *compression* result.

5.1 Infinitary rewriting and other preliminary material

The work on infinitary rewriting we describe in this chapter resorts to several definitions and properties. The present section describes this preliminary material.

The subject of Section 5.1.1 is the theory of countable ordinals; we present some definitions and results which are particularly relevant to the extension of the proof term model to infinitary rewriting.

In Sections 5.1.2 to 5.1.4, we give the definitions of the main concepts of infinitary rewriting we use in the following. In Section 5.1.2, we deal with the definition of *infinitary term*, and of the related notions of *position* and *context*. Some basic results are given as well. In Section 5.1.3, we study the extension, to the infinitary setting, of the notion of *substitution*, verifying that it enjoys some expected properties. Section 5.1.4 is devoted to the definition of *infinitary (first-order) term rewriting system*, and to formalise the notions of *reduction step* and *reduction sequence*.

Finally, in Section 5.1.5 we introduce the notion of *pattern*, and Section 5.1.6 includes some results about infinitary rewriting which are needed in the following.

5.1.1 Countable ordinals

We do not give a general presentation of the theory of ordinals. The general references we use for this subject are [Sup60, Sie65, For03].

In order to deal with infinitary composition, we will need to obtain the sum of a sequence including ω ordinals. Thus we will resort to the following definition, cfr. [Sup60] Dfn. 6 pg. 216.

Definition 5.1.1 (Ordinal infinitary sum). *Let $\langle \alpha_i \rangle_{i<\omega}$ be a sequence of ordinals. We define the sum of $\langle \alpha_i \rangle_{i<\omega}$ as follows:*

$$\sum_{i<\omega} \alpha_i := sup(\{\alpha_0 + \alpha_1 + \ldots + \alpha_{n-1} + \alpha_n \ / \ n < \omega\})$$

The sum of ω ordinals, in the way it was just defined, enjoys the following important property.

Lemma 5.1.2. *Let $\langle \alpha_i \rangle_{i<\omega}$ be a sequence of ordinals, and β an ordinal such that $\beta < \sum_{i<\omega} \alpha_i$. Then there exist a unique $k < \omega$ and an ordinal γ such that $\beta = \alpha_0 + \ldots + \alpha_{k-1} + \gamma$ and $\gamma < \alpha_k$.*

Proof. This is an easy consequence of some properties of ordinals. Namely, $\beta < \sum_{i<\omega} \alpha_i$ implies that the set $\{k < \omega \ / \ \beta < \alpha_0 + \ldots + \alpha_k\}$ is nonempty; we take n as the minimum of this set. Then $\alpha_0 + \ldots + \alpha_{n-1} \leqslant \beta < (\alpha_0 + \ldots + \alpha_{n-1}) + \alpha_n$. Basic properties of ordinals entail the existence and uniqueness of an ordinal γ verifying $(\alpha_0 + \ldots + \alpha_{n-1}) + \gamma = \beta$, and also that $\gamma < \alpha_n$. Thus we conclude. □

Finally, the property of ω-cofinality of countable ordinals, cfr. [For03] Remark 73 pg. 169, is needed in some proofs along this chapter. Specifically, we resort to the following consequence of this property.

Proposition 5.1.3. *Let α be a limit countable ordinal. Then there exists an increasing sequence of ordinals $\langle \alpha_i \rangle_{i<\omega}$ such that $0 < \alpha_i < \alpha$ for all $i < \omega$, and $\alpha = \sum_{i<\omega} \alpha_i$.*

5.1.2 Positions, terms, contexts

We consider the usual definition of the notion of *position*.

Definition 5.1.4 (Position, depth of a position). *A* position *is a finite sequence of* $\mathbb{N}_{>0}$. *The empty sequence is denoted by the symbol* ϵ. *The* depth *of a position* p*, notation* $|p|$*, is defined as its length; observe that* $|\epsilon| = 0$.

Definition 5.1.5 (Concatenation of positions). *Let* p, q *be positions. Then we define* $p \cdot q$*, the concatenation of* p *and* q*, as follows:* $\epsilon \cdot q := q$ *and* $(ip) \cdot q := i(p \cdot q)$*. Moreover, given* P, Q *sets of positions, then we define also* $P \cdot q := \{p \cdot q \mathrel{/} p \in P\}$ *and* $p \cdot Q := \{p \cdot q \mathrel{/} q \in Q\}$.

We will omit the dot to denote concatenation, i.e. we will write pq, pQ, Pq *instead of* $p \cdot q, p \cdot Q, P \cdot q$ *wherever no confusion arises.*

Following e.g. [Cou83], the definition of infinitary term is based on the notion of *tree domain*. The notion of signature can be defined exactly as for finitary term rewriting, cfr. Dfn. 2.2.1.

Definition 5.1.6 (Tree domain). *A* tree domain *is any set of positions* P *satisfying the following conditions (*p, q *positions;* $i, j \in \mathbb{N}_{>0}$*):* $P \neq \varnothing$*;* P *is prefix closed, i.e.* $pq \in P$ *implies* $p \in P$ *(particularly,* $\epsilon \in P$*); if* $pj \in P$ *and* $1 \leqslant i < j$*, then* $pi \in P$.

Definition 5.1.7 (Term, positions of a term, symbol at a position, sets of finitary and infinitary terms). *A* term *over a signature* Σ *and a countable set of variables* $\texttt{Var}$ *is any pair* $\langle P, F \rangle$*, such that* P *is a tree domain,* $F : P \to \Sigma \cup \texttt{Var}$*, and the following condition holds: if* $p \in P$ *and* $F(p) = h$*, then* $pi \in P$ *iff* $i \leqslant ar(h)$*, where we consider* $ar(x) = 0$ *if* $x \in \texttt{Var}$.[2]

If $t = \langle P, F \rangle$ *is a term, we will denote* P *by* $\texttt{Pos}(t)$*, and* F *just by* t*; therefore, we will write* $t(p)$ *to denote* $F(p)$.

A term is finite *iff its tree domain is, otherwise it is* infinite.

Given a signature Σ *and a countable set of variables* $\texttt{Var}$*, the set of* finitary terms *over* Σ*, notation* $Ter(\Sigma, \texttt{Var})$*, is the set of finite terms over* Σ*; and the set of* infinitary terms *over* Σ*, notation* $Ter^{\infty}(\Sigma, \texttt{Var})$*, is the set of finite or infinite terms over* Σ.

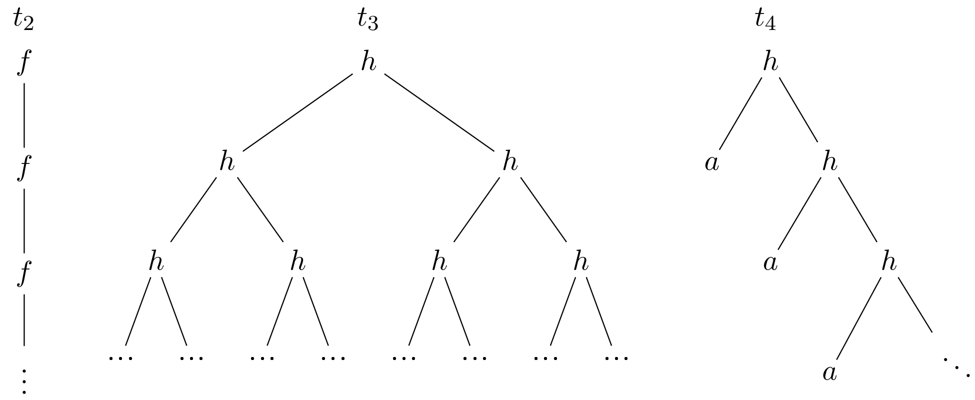
We will often drop the set of variables, writing just $Ter(\Sigma)$ *or* $Ter^{\infty}(\Sigma)$.

We will name **head symbol** of a term t the symbol $t(\epsilon)$. The name **root symbol** will be used as well.

We give some examples of terms, according to Dfn. 5.1.7. We use the symbols $h/2$, $f/1$, $a/0$, $b/0$. The term $t_1 = h(f(a), b)$ is described formally as $\langle P_1, F_1 \rangle$, where $P_1 = \{\epsilon, 1, 11, 2\}$ and $F_1 = \{\epsilon \to h, 1 \to f, 11 \to a, 2 \to b\}$. This is a finite term, because

[2]in some texts, e.g. [Cou83] and [Gal86], a term is defined just as a function from positions to symbols; the set of positions is implicitly determined by being the domain of the function. We prefer to explicitly include the set of positions in the definition.

so is P_1. We have e.g. $t_1(1) = f$ and $t_1(2) = b$. We show graphically three infinite terms:



The term t_2 corresponds to the idea described informally as $f(f(f(\ldots)))$. Formally, we define $t_2 = \langle P_2, F_2 \rangle$, where $P_2 = \{1^n \;/\; n < \omega\}$ and $F_2(1^n) = f$ for all $n < \omega$. In turn, $t_3 = \langle P_3, F_3 \rangle$, where P_3 is the set of all finite sequences built using the numbers 1 and 2, and $F_3(p) = h$ for all $p \in P_3$. Finally, $t_4 = \langle P_4, F_4 \rangle$, where $P_4 = \bigcup_{n<\omega} \{2^n, 2^n 1\}$, and for all $n < \omega$, $F_4(2^n) = h$ and $F_4(2^n 1) = a$.

Notation 5.1.8 (Intuitive notation for terms)**.** *An alternative notation will be often used for terms in $Ter^\infty(\Sigma, \texttt{Var})$: if $x \in \texttt{Var}$ and $f/n \in \Sigma$, then we will write*

- *x for $\langle \{\epsilon\}, F \rangle$ where $F(\epsilon) = x$, and*
- *$f(t_1, \ldots, t_n)$ for $\langle P, F \rangle$, where $P = \{\epsilon\} \cup \bigcup_{1 \leqslant i \leqslant n} \{ip \;/\; p \in \texttt{Pos}(t_i)\}$, $F(\epsilon) = f$, and $F(ip) = t_i(p)$.*

We will use $t \in \texttt{Var}$ as shorthand notation for $t = \langle \{\epsilon\}, F \rangle$, $F(\epsilon) = x$, and $x \in \texttt{Var}$.
If $f/1 \in \Sigma$, then we will write f^ω for the term $t = f(f(f(\ldots)))$, whose formal definition is described above.[3]

We observe that any term comprised in Dfn. 5.1.7 can be described using Notation 5.1.8.

Proposition 5.1.9. *Let $t \in Ter^\infty(\Sigma, \texttt{Var})$. Then either $t = x$ or $t = f(t_1, \ldots, t_n)$ where $f/n \in \Sigma$ and $t_i \in Ter^\infty(\Sigma, \texttt{Var})$ for all $i \leqslant n$; cfr. Notation 5.1.8.*

Proof. Dfn.. 5.1.6 implies that $\epsilon \in \texttt{Pos}(t)$.

Assume $t(\epsilon) = x \in \texttt{Var}$. Moreover, assume for contradiction the existence of some $p \in \texttt{Pos}(t)$ such that $p \neq \epsilon$. In that case there should be some $n \in \mathbb{N}$ being the minimum of the depths of such positions, i.e. $n = min(|p| \;/\; p \in \texttt{Pos}(t) \wedge p \neq \epsilon)$. Observe that $n = 1$ would imply the existence of some $i \in \mathbb{N}$ verifying $i \in \texttt{Pos}(t)$, contradicting Dfn. 5.1.7 since we consider $ar(x) = 0$. In turn, $n > 1$ would entail $p = p'i \in \texttt{Pos}(t)$ for some p verifying $|p| = n$ and $|p'| > 0$, implying $p' \in \texttt{Pos}(t)$ by Dfn. 5.1.6, thus contradicting minimality of n. Consequently, $\texttt{Pos}(t) = \{\epsilon\}$, hence $t = x$.

Assume $t(\epsilon) = f \in \Sigma$. For each $i \in \mathbb{N}$ we define $P_i := \{p \;/\; ip \in \texttt{Pos}(t)\}$, and $F_i : P_i \to \Sigma \cup \texttt{Var}$ such that $F_i(p) := t(ip)$. If $i \leqslant ar(f)$, then $P_i \neq \varnothing$ since $\epsilon \in P_i$. Moreover, $\texttt{Pos}(t)$ being a tree domain implies immediately that P_i enjoys the remaining

[3]This convention could generalise to any $f/n \in \Sigma$, by defining $f^\omega = \langle P, F \rangle$ where P is the set of all the sequences that can be built using the numbers $\{1, 2, \ldots, n\}$, and $F(p) := f$ for all $p \in P$. Roughly speaking, f^ω would be defined as the infinite tree all filled with f.

conditions in Dfn. 5.1.6; and also the condition on F_i described in Dfn. 5.1.7 stems immediately from the fact that t is a term. Therefore, $t_i := \langle P_i, F_i \rangle$ is a term. On the other hand, $i > ar(f)$ implies that $P_i = \varnothing$, thus $\texttt{Pos}(t) = \{\epsilon\} \cup \bigcup_{1 \leqslant i \leqslant ar(f)} \{ip \; / \; p \in P_i\}$. We conclude by observing that $t = f(t_1, \ldots, t_n)$. □

Some notions related to terms

A number of basic definitions pertaining to first-order term rewriting extend to the infinitary setting; some expected properties are preserved. For the finitary counterparts of these definitons and properties, cfr. e.g. [BN98], Section 3.1.

Definition 5.1.10 (Occurrence)**.** *Let t be a (either finite or infinite) term over Σ and $a \in \Sigma \cup \texttt{Var}$. An* occurrence *of a in t is a position $p \in \texttt{Pos}(t)$ such that $t(p) = a$. We define $\texttt{Occ}_a(t)$ as the set of occurrences of a in t.*

A symbol $a \in \Sigma \cup \texttt{Var}$ occurs in *a term t iff $\texttt{Occ}_a(t) \neq \varnothing$, i.e. iff there is at least one occurrence of a in t; a occurs exactly $n \in \mathbb{N}$ times in t iff $|\texttt{Occ}_a(t)| = n$, where $|S|$ denotes the cardinal of any set S.*

Definition 5.1.11 (Closed term, linear term)**.** *A term t is said to be* closed *iff it includes no occurrences of variables; it is said to be* linear *iff no variable occurs in it more than once.*

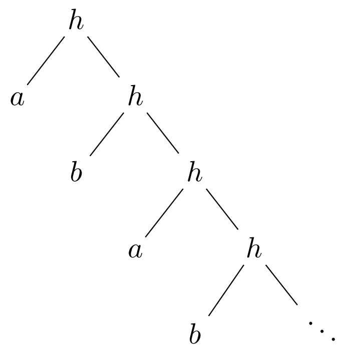
For example, the symbol f has two occurrences in the term $t = h(f(a), g(f(b)))$; more precisely, $\texttt{Occ}_f(t) = \{1, 21\}$. The symbols h, a, b and g also occur in t, while e.g. the symbol c does not. Obseve that no variable occurs in t, so that it is a closed term.

Definition 5.1.12 (Subterm at a position)**.** *Let $t = \langle P, F \rangle$ be a term, and $p \in P$. We define the* subterm *of t at position p, notation $t|_p$, as $\langle P|_p, F|_p \rangle$, where $P|_p$ and $F|_p$ are the* projections *of P and F over p respectively; i.e., $P|_p := \{q \; / \; pq \in P\}$ and $F|_p : P|_p \to \Sigma \cup \texttt{Var}$ such that $F|_p \; (q) := F(pq)$.*

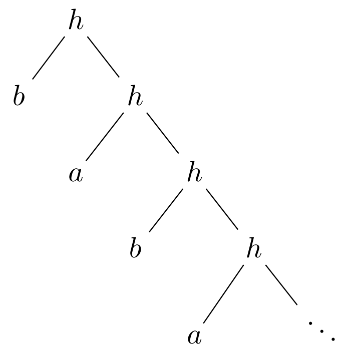
A simple example shows that Dfn. 5.1.12 extends the usual definition of subterm at a position. Let $t = h(h(f(a), g(b)), g(h(b, a)))$, so that $t = \langle P, F \rangle$ where $P = \{\epsilon, 1, 11, 111, 12, 121, 2, 21, 211, 212\}$ and $F = \{\epsilon \to h, 1 \to h, 11 \to f, 111 \to a, 12 \to g, 121 \to b, 2 \to g, 21 \to h, 211 \to b, 212 \to a\}$. Then $P|_2 = \{\epsilon, 1, 11, 12\}$. In turn, $F|_2 \; (1) = F(21) = h$. Analogously, we obtain $F|_2 \; (\epsilon) = g$, $F|_2 \; (11) = b$ and $F|_2 \; (12) = a$. Hence, $t|_2 = \langle \{\epsilon, 1, 11, 12\}, \{\epsilon \to g, 1 \to h, 11 \to b, 12 \to a\} \rangle$, that is, the term $g(h(b, a))$.

An example involving infinitary terms follows. Let t_1 and t_2 be the following terms:

$t_1 = \langle P_1, F_1 \rangle$



$t_2 = \langle P_2, F_2 \rangle$



so that $P_1 = P_2 = \bigcup_{n<\omega}\{2^n, 2^n 1\}$, $F_1(2^n) = F_2(2^n) = h$, $F_1(2^{2n}1) = F_2(2^{2n+1}1) = a$ and $F_1(2^{2n+1}1) = F_2(2^{2n}1) = b$. Then $t_1|_1 = a$ and $t_2|_1 = b$, as expected. In turn, it is not difficult to grasp that $P_1|_2 = P_1 = P_2$, $F_1|_2\ (2^n) = F_1(2^{n+1}) = h$, $F_1|_2\ (2^{2n}1) = F_1(2^{2n+1}1) = b$, and $F_1|_2\ (2^{2n+1}1) = F_1(2^{2n}1) = a$. Therefore, $t_1|_2 = t_2$. Analogously, we obtain $t_2|_2 = t_1$. These results coincide with what can be easily observed in the graphical description of t_1 and t_2.

Dfn. 5.1.12 allow a straightforward and direct (i.e. non-inductive) proof of a basic result about subterms. Namely

Lemma 5.1.13. $t|_{pq} = (t|_p)|_q$.

Proof. If we call $\langle P, F\rangle := t|_{pq}$ and $\langle P', F'\rangle := (t|_p)|_q$, then Dfn. 5.1.12 yields

$$\begin{array}{rclcrcl} P & = & \{r \ / \ pqr \in \texttt{Pos}(t)\} & \quad & P' & = & \{r \ / \ qr \in \texttt{Pos}(t|_p)\} \\ F(r) & = & t(pqr) & & F'(r) & = & t|_p\ (qr) = t(pqr) \end{array}$$

We conclude by observing that $pqr \in \texttt{Pos}(t)$ iff $qr \in \texttt{Pos}(t|_p)$. □

Particularly, if $t = f(t_1, \ldots, t_n)$, then $t|_{ip} = t_i|_p$; cfr. Notation 5.1.8.

Definition 5.1.14 (Replacement at a position). *Let t and u be terms, and $p \in \texttt{Pos}(t)$. We define the* replacement *of t under position p with u, notation $t[u]_p$, as $\langle P', F'\rangle$ such that $P' := \{q \in \texttt{Pos}(t) \ / \ p \nleq q\} \cup \{pq \ / \ q \in \texttt{Pos}(u)\}$ and*

$$F'(q) := \begin{cases} t(q) & \text{iff} \ \ p \nleq q \\ u(q') & \text{iff} \ \ q = pq' \end{cases}.$$

We state and prove some basic properties about replacement. It is worth mentioning that the definition of infinitary term we give in Dfn. 5.1.7 is of a different nature from the definitions of (finitary) term given in [BKdV03] (Dfn. 2.1.2, page 26) and [BN98] (Dfn. 3.1.2, page 35), so that it is necessary to verify these properties.

Lemma 5.1.15. *Let $t = f(t_1, \ldots, t_n)$ and u be terms, and $p \in \texttt{Pos}(t_i)$. Then $t[u]_{ip} = f(t_1, \ldots, t_i[u]_p, \ldots, t_n)$.*

Proof. Let us call $t' = \langle P', F'\rangle := f(t_1, \ldots, t_n)[u]_{ip}$ and $t'' = \langle P'', F''\rangle := f(t_1, \ldots, t_i[u]_p, \ldots, t_n)$.

By joining Notation 5.1.8 and Dfn. 5.1.14 we obtain $P' = \{\epsilon\} \cup \{jq \ / \ q \in \texttt{Pos}(t_j) \land j \neq i\} \cup \{iq' \ / \ q' \in \texttt{Pos}(t_i) \land p \nleq q'\} \cup \{ipq \ / \ q \in \texttt{Pos}(u)\}$. It is straightforward to verify that $P' = P''$; particularly, notice that $\texttt{Pos}(t_i[u]_p) = \{q' \ / \ q' \in \texttt{Pos}(t_i) \land p \nleq q'\} \cup \{pq \ / \ q \in \texttt{Pos}(u)\}$.

Let us compare $F'(p)$ and $F''(p)$, for any $p \in P' = P''$. $F'(\epsilon) = F''(\epsilon) = f$. If $j \neq i$ then $ip \nleq jq$, then $F'(jq) = F''(jq) = t_j(q)$. If $p \nleq q'$, then $F'(iq') = F(iq') = t_i(q')$, and $F''(iq') = t_i[u]_p(q') = t_i(q')$. Finally, if $q = pq'$, then $F'(iq) = u(q')$ and $F''(iq) = t_i[u]_p(pq') = u(q')$. Thus we conclude. □

Lemma 5.1.16. *Let t and u be terms and $pq \in \texttt{Pos}(t)$. Then $t[u]_{pq} = t[t|_p\ [u]_q]_p$.*

Proof. By induction on p.

If $p = \epsilon$, then both $t[u]_{pq}$ and $t[t|_p\ [u]_q]_p$ are equal to $t[u]_q$.

Assume that $p = ip'$, in this case $t = g(t_1, \ldots, t_n)$. Lem. 5.1.15 implies that $t[u]_{pq} = t[u]_{ip'q} = g(t_1, \ldots, t_i[u]_{p'q}, \ldots t_n)$ and also $t[t|_p\ [u]_q]_p = t[t|_{ip'}\ [u]_q]_{ip'} = g(t_1, \ldots, t_i[t_i|_{p'}\ [u]_q]_{p'}, \ldots, t_n)$. We conclude by IH on p', t_i and u. □

Lemma 5.1.17. *Let t, s be terms and $p, q \in \texttt{Pos}(t)$ such that $p \parallel q$. Then $(t[s]_q)|_p = t|_p$.*

Proof. Say $t = \langle P, F\rangle$, $t[s]_q = \langle P', F'\rangle$, $t|_p = \langle P_p, F_p\rangle$, and $(t[s]_q)|_p = \langle P'_p, F'_p\rangle$. We prove $P_p = P'_p$ by double inclusion.

$\subseteq$) Let $p' \in P_p$, so that $pp' \in P$. Observe that $p \parallel q$ implies $pp' \parallel q$, so that $q \nleqslant pp'$, implying $pp' \in P'$, and therefore $p' \in P'_p$.

$\supseteq$) Let $p' \in P'_p$, so that $pp' \in P'$. We have already verified $q \nleqslant pp'$, so that the only valid option w.r.t. Dfn. 5.1.14 is $pp' \in P$, implying $p' \in P_p$.

Let $p' \in P'_p = P_p$, so that $pp' \in P \cap P'$ and $q \nleqslant pp'$. Dfn. 5.1.12 implies $F'_p(p') = F'(pp')$ and $F_p(p') = F(pp')$. In turn, Dfn. 5.1.14 yields $F'(pp') = F(pp')$, since $q \nleqslant pp'$. Consequently $F_p = F'_p$. Thus we conclude. □

Contexts

The notion of *context* also extends to infinitary terms as expected, provided that the contexts we deal with in this chapter include only a *finite number of holes*. In some situations, particularly in the definition of rewriting rules, variable occurrences play a role similar to that of the holes in a context.

Definition 5.1.18 (Context, one-hole context)**.** *A* context *over Σ is a term (either finite or infinite) over $\Sigma \cup \{\Box/0\}$. A* one-hole context *is a context in which the symbol $\Box$ occurs exactly once.*

Definition 5.1.19 (Position of a variable (hole) occurrence in a term (context))**.** *Let t be a term. Then we define $\texttt{VOccs}(t) := \{p \;/\; t(p) \in \texttt{Var}\}$. Given a term t, if $|\texttt{VOccs}(t)| = n \in \mathbb{N}$, then for any i such that $1 \leqslant i \leqslant n$ we define $\texttt{Vpos}(t, i)$, the i-th variable occurrence in t, as the i-th element of the set $\texttt{VOccs}(t)$, considering the order given by $p < q$ iff $|p| < |q|$ or $|p| = |q|$, $p = rip'$, $q = rjq'$, $i < j$.*

Analogously, if C is a context including a finite number of occurrences of the symbol $\Box$, then we define $\texttt{Bpos}(C, i)$ as the i-th element of $\texttt{Occ}_\Box(C)$, considering the order just described.

Definition 5.1.20 (Context replacement)**.** *Let C be a context including exactly n occurrences of the symbol $\Box$, and $t_1, \ldots, t_n$ terms. We define the replacement of C using $t_1, \ldots, t_n$ as $C[t_1, \ldots, t_n] := \langle P, F\rangle$, where*
$P := \{p \in \texttt{Pos}(C) \;/\; C(p) \neq \Box\} \cup \bigcup_i \{\texttt{Bpos}(C, i) \cdot p \;/\; p \in \texttt{Pos}(t_i)\}$,
and $F'(p) := \begin{cases} C(p) & \text{iff} \quad C(p) \neq \Box \\ t_i(q) & \text{iff} \quad p = \texttt{Bpos}(C, i) \cdot q \end{cases}$

It is easy to verify an expected result about context replacement, namely:

Lemma 5.1.21. $C[t_1, \ldots, t_n]|_{\texttt{Bpos}(C,i)\cdot p} = t_i|_p$

Proof. Immediate from Dfn. 5.1.20. □

Distance between terms, equality, metric space of terms

The following notion of *distance* is used to ascertain the equality of terms.

Definition 5.1.22 (Distance between terms, cfr. [BKdV03] p. 670)**.** *Let t, u be terms. We define the* distance *between t and u, notation* $\texttt{dist}(t, u)$*, as follows:*

- *0 iff $t = u$, and*
- *2^{-k} otherwise, where k is the length of the shortest position at which the two terms differ; i.e. $k = |p|$* s.t. *p is minimal for $p \in \texttt{Pos}(t) \cup \texttt{Pos}(u)$ and either $p \notin \texttt{Pos}(t) \cap \texttt{Pos}(u)$ or $t(p) \neq u(p)$.*

Remark 5.1.23 (Equality criterion for terms)**.** *Dfn. 5.1.22 implies that, given two terms t and u, obtaining $\texttt{dist}(t, u) < 2^{-k}$ for all $k < \omega$ is a sufficient condition to conclude $t = u$. In turn, to check $\texttt{dist}(t, u) < 2^{-k}$ it is enough to verify, for any position p, that $|p| \leqslant k$ and $p \in \texttt{Pos}(t) \cup \texttt{Pos}(u)$ entails $p \in \texttt{Pos}(t) \cap \texttt{Pos}(u)$ and $t(p) = t(u)$.*

We give some examples of Dfn. 5.1.22, involving the terms $t_1 = h(f(f(a)), g(g(g(b))))$, $t_2 = h(f(f(a)), g(b))$ and $t_3 = h(f^\omega, g^\omega)$. We obtain $\texttt{dist}(t_1, t_2) = 2^{-2}$, because $t_1(21) = g \neq b = t_2(21)$, and for all $p \in \texttt{Pos}(t_1) \cup \texttt{Pos}(t_2)$ such that $|p| < 2$, that is, for $p = \epsilon, 1, 2$, we have $t_1(p) = t_2(p)$. Analogously, we obtain $\texttt{dist}(t_1, t_3) = 2^{-3}$, since $t_1(111) = a \neq f = t_3(111)$, and $\texttt{dist}(t_2, t_3) = 2^{-2}$, since $t_2(21) = b \neq g = t_3(21)$.

The notion of distance given by Dfn. 5.1.22 allows to define the *limit* of an infinite sequence of terms.

Definition 5.1.24 (Limit of a sequence of terms)**.** *Let $< t_i >_{i<\alpha}$ a sequence of terms where α is a countable limit ordinal. We say that the sequence $< t_i >$ has the term t as its limit (notation $\lim_{i\to\alpha} t_i = t$) iff the following limit condition holds: for any $p \in \mathbb{N}$ there exists $k_p < \alpha$ such that for all j satisfying $k_p < j < \alpha$, $\texttt{dist}(t_j, t) < 2^{-p}$.*

E.g., let $t_i = f^i(a)$. To conclude that $\lim_{i\to\omega} t_i = f^\omega$, it suffices to observe that $\texttt{dist}(t_i, f^\omega) = 2^{-i}$, for all $i < \omega$.

Dfn. 5.1.22 yields a *metric*, which can be applied to both $Ter(\Sigma)$ and $Ter^\infty(\Sigma)$. The set $Ter^\infty(\Sigma)$ turns out to be isomorphic to the metric completion of $Ter(\Sigma)$ w.r.t. this metric, cfr. e.g. [AN80, KKSdV95] and [BKdV03] pp. 670/671; it is therefore metric-complete on $Ter(\Sigma)$, and also on $Ter^\infty(\Sigma)$ itself. Consequently, for any Cauchy-convergent sequence of terms, a term exists which is the limit of that sequence.

The set $Ter^\infty(\Sigma)$ forms, moreover, an *ultrametric* space along with the given metric. Formally:

Lemma 5.1.25. *Let t, u, w be terms. Then $\texttt{dist}(t, w) \leqslant max(\texttt{dist}(t, u), \texttt{dist}(u, w))$.*

Proof. If $t = u = w$, then all distances are 0. Otherwise, we proceed by induction on k where $max(\texttt{dist}(t, u), \texttt{dist}(u, w)) = 2^{-k}$. If $k = 0$ we conclude immediately since the distance between any pair of terms cannot be greater than one. Assume $k = k' + 1$. Then $\texttt{dist}(t, u) < 2^{-k'}$, implying that for any position p such that $|p| \leqslant k'$, it is easy to verify that $p \in \texttt{Pos}(t)$ iff $p \in \texttt{Pos}(u)$, and moreover, $p \in \texttt{Pos}(t)$ implies $t(p) = u(p)$. On the other hand, the same properties hold for u w.r.t. w, since $\texttt{dist}(u, w) < 2^{-k'}$. Hence $\texttt{dist}(t, w) \leqslant 2^{-k}$, thus we conclude. □

The distance between a term and the result of a replacement on that term is limited by the depth of the position corresponding to the replacement. Namely:

Lemma 5.1.26. *Let t, s be terms and $p \in \texttt{Pos}(t)$. Then $\texttt{dist}(t, t[s]_p) \leqslant 2^{-|p|}$.*

Proof. We proceed by induction on p. If $p = \epsilon$ then we conclude immediately since $\mathtt{dist}(t, u) \leqslant 2^0 = 1$ for any term u. Otherwise, i.e. if $p = ip'$, observe that $ip' \in \mathtt{Pos}(t)$ implies $t = f(t_1, \ldots, t_i, \ldots, t_m)$. Then $t[s]_p = f(t_1, \ldots, t_i[s]_{p'}, \ldots, t_m)$, cfr. Lem. 5.1.15, implying $\mathtt{dist}(t, t[s]_p) = \frac{1}{2} * \mathtt{dist}(t_i, t_i[s]_{p'})$. In turn, IH yields $\mathtt{dist}(t_i, t_i[s]_{p'}) \leqslant 2^{-|p'|}$. Therefore, easy exponent arithmetics recalling $|p| = |p'| + 1$ suffices to conclude. □

5.1.3 Substitutions

The definition of *substitution* extends, in a natural way, from finitary to infinitary terms.

Definition 5.1.27 (Substitution)**.** *Given a set of variables* $\mathtt{Var}$ *and a signature* Σ*, a* substitution *is a function* $\sigma : \mathtt{Var} \to Ter^\infty(\Sigma, \mathtt{Var})$ *where* $\sigma(x) = x$ *except for a finite subset of* $\mathtt{Var}$*.*

Any substitution is extended into a function, bearing the same name σ*, where* $\sigma : Ter^\infty(\Sigma, \mathtt{Var}) \to Ter^\infty(\Sigma, \mathtt{Var})$*, defined as follows:* $\sigma t := \langle P, F \rangle$ *where*
$P = \{p \in \mathtt{Pos}(t) \;/\; t(p) \notin \mathtt{Var}\} \cup \{pq \;/\; t(p) = x \in \mathtt{Var} \wedge q \in \mathtt{Pos}(\sigma x)\}$ *and*

$$F(p) = \begin{cases} t(p) & \text{iff} \quad p \in \mathtt{Pos}(t) \wedge t(p) \notin \mathtt{Var} \\ \sigma x(q') & \text{iff} \quad p = qq' \wedge t(q) = x \in \mathtt{Var} \end{cases}$$

For finitary terms, the extension of the domain of a substitution from variables to terms can be defined by resorting to the concept of Σ-*algebra*; cfr. [BN98] Chapter 3. Given a signature Σ, we can define a Σ-algebra whose carrier set is $Ter(\Sigma, \mathtt{Var})$, which we will denote by $Ter(\Sigma, \mathtt{Var})$ as well. For any $f/n \in \Sigma$, the corresponding function is defined simply as follows:

$$f^{Ter(\Sigma,\mathtt{Var})}(t_1, \ldots, t_n) := f(t_1, \ldots, t_n)$$

This Σ-algebra is *generated* by $\mathtt{Var}$. Then the result of uniqueness of homomorphisms for a Σ-algebra generated by a set, given the values for the generator set, cfr. [BN98] Lem. 3.3.1, allows to define the extension of the substitution σ to terms as the *only* endomorphism on (the Σ-algebra whose carrier set is) $Ter(\Sigma)$ coinciding with σ for all the variables.

We can consider a Σ-algebra having as carrier set $Ter^\infty(\Sigma)$, defined as we just did for $Ter(\Sigma)$; cfr. Notation 5.1.8. The extension of any substitution σ to $Ter^\infty(\Sigma)$, as given in Dfn. 5.1.27, is an endomorphism on this Σ-algebra.

Lemma 5.1.28. *Let* σ *be a substitution defined only on variables, and* $\hat{\sigma}$ *the corresponding extension to* $Ter^\infty(\Sigma)$*. Then* $\hat{\sigma}$ *is an endomorphism on* $Ter^\infty(\Sigma, \mathtt{Var})$ *which extends* σ*.*

Proof. It is enough to show that $\hat{\sigma}(f(t_1, \ldots, t_n)) = f(\hat{\sigma}(t_1), \ldots, \hat{\sigma}(t_n)))$; cfr. Prop. 5.1.9; let us call these terms $t' = \langle P', F' \rangle$ and $t'' = \langle P'', F'' \rangle$ respectively.

By applying notation 5.1.8 and Dfn. 5.1.27, we obtain

$$\begin{array}{rcll} P' & = & \{\epsilon\} \cup \bigcup_i (\{ip \;/\; p \in \mathtt{Pos}(t_i) \wedge t_i(p) \notin \mathtt{Var}\} \cup & \\ & & \qquad \{ipq \;/\; t_i(p) = x \in \mathtt{Var} \wedge q \in \mathtt{Pos}(\sigma x)\}) & \\ F'(\epsilon) & = & f & \\ F'(ip) & = & t_i(p) & \text{if } p \in \mathtt{Pos}(t_i) \wedge t_i(p) \notin \mathtt{Var} \\ F'(ipq) & = & \sigma x(q) & \text{if } t_i(p) = x \in \mathtt{Var} \wedge q \in \mathtt{Pos}(\sigma x) \end{array}$$

An analogous analysis for P'' and F'' is enough to conclude. □

Nonetheless, we cannot use the mentioned result on uniqueness of homomorphisms on generated Σ-algebras, given the values for the generator set, to assert that $\widehat{\sigma}$ is the only endomorphism on $Ter^{\infty}(\Sigma, \mathtt{Var})$ which extends σ. The reason is that $Ter^{\infty}(\Sigma, \mathtt{Var})$ is not generated by $\mathtt{Var}$: notice that the Σ-subalgebra generated by $\mathtt{Var}$ for $Ter^{\infty}(\Sigma, \mathtt{Var})$ is exactly $Ter(\Sigma, \mathtt{Var})$.

Fortunately, an analogous uniqueness result can be proved for endomorphisms on $Ter^{\infty}(\Sigma, \mathtt{Var})$.

Proposition 5.1.29. *Let Σ be a signature, and ϕ, ψ two endomorphisms on the Σ-algebra $Ter^{\infty}(\Sigma, \mathtt{Var})$ which coincide on $\mathtt{Var}$. Then $\phi = \psi$.*

Proof. We will prove the following statement, which entails the desired result (i.e. that for any term t, $\psi(t) = \phi(t)$): for any $k < \omega$, given a term t and a position p such that $|p| \leqslant k$ and $p \in \mathtt{Pos}(\psi(t)) \cup \mathtt{Pos}(\phi(t))$, then $\psi(t)(p) = \phi(t)(p)$. Cfr. comment following Dfn. 5.1.22.

We proceed by induction on k. There is one case which does not need to resort to the inductive argument: if $t \in \mathtt{Var}$, then $\psi(t) = \phi(t)$ since hypotheses assert that these functions coincide on $\mathtt{Var}$.

Thus assume $t = f(t_1, \ldots, t_m)$; cfr. Prop. 5.1.9. In this case hypotheses entail $\psi(t) = f(\psi(t_1), \ldots, \psi(t_m))$ and $\phi(t) = f(\phi(t_1), \ldots, \phi(t_m))$. If $k = 0$, then $|p| \leqslant k$ implies $p = \epsilon$, hence it is enough to observe that $\psi(t)(\epsilon) = \phi(t)(\epsilon) = f$. Assume $k = k' + 1$. If $|p| \leqslant k'$ then applying IH on k' w.r.t. t and q suffices to conclude. If $|p| = k$, then $p = iq$ (recall $k > 0$) where $|q| = k'$ and $q \in \mathtt{Pos}(\psi(t_i)) \cup \mathtt{Pos}(\phi(t_i))$. Therefore we can apply IH on k' w.r.t. t_i and q, obtaining $\psi(t_i)(q) = \phi(t_i)(q)$. Thus we conclude by observing $\psi(t)(p) = \psi(t_i)(q)$ and analogously for ϕ. □

Consequently, we can assert that $\widehat{\sigma}$ is the only endomorphism on $Ter^{\infty}(\Sigma, \mathtt{Var})$ which extends σ, as desired.

5.1.4 Rewriting: TRS, step, reduction sequence

The definition of *infinitary term rewriting system* is the natural extension of its finitary counterpart, cfr. Dfn. 2.2.2, provided that the left-hand side of rewriting rules is required to be a finite term.

Definition 5.1.30 (Rewrite rule, term rewriting system)**.** *Assuming a set of variables* $\mathtt{Var}$ *and given a signature Σ, a* rewrite rule *(just* rule *if no confusion arises) over Σ is a pair of terms $\langle l, r\rangle$ satisfying the following conditions: l is a finite term, $l \notin \mathtt{Var}$, and each variable occurring in r occurs also in l. Notation for a rewrite rule: $l \to r$, also $\mu : l \to r$ if assigning explicit names to rules is desirable. The terms l and r, respectively, are the* left-hand side *and* right-hand *side,* lhs *and* rhs *for short, of the rule $l \to r$.*

A term rewriting system *(shorthand TRS) is a pair $T = \langle \Sigma, R\rangle$, where Σ is a signature and R is a set of rules over Σ.*

If the right-hand sides of all the rules are finite terms, then T can be considered as a TRS over either $Ter(\Sigma)$ or $Ter^{\infty}(\Sigma)$; otherwise, only the infinitary interpretation is valid. In either case, a TRS over $Ter^{\infty}(\Sigma)$ is known as a infinitary TRS*, or iTRS for short.*

We say that a rewrite rule $\mu : l \to r$ is *collapsing* iff $r \in \mathtt{Var}$.

We define that a TRS is *left-linear* iff for any l left-hand side of a rule, and for any x variable, x occurs in l at most once. In this chapter, we study left-linear iTRSs only. A TRS T is *orthogonal* iff it is left-linear and there is no term t such that $t = \sigma_1 l_1$ and $t|_p = \sigma_2 l_2$, where l_1 and l_2 are left-hand sides of rules in T, $l_1(p)$ is a function symbol (i.e. it is not a variable), and either $p \neq \epsilon$ or the rules corresponding to l_1 and l_2 do not coincide.

Some examples of left-hand sides of rules leading to non-orthogonal TRSs follow. No TRS including a rule whose left-hand side is $f(g(x))$ and another having as left-hand side either $g(x)$ or $g(h(x))$, is orthogonal: $t = f(g(h(a)))$ is a counterexample for the corresponding condition. Also, no TRS including rules whose left-hand sides are $h(f(x), y)$ and $h(x, g(y))$ is orthogonal, a counterexample is $t = h(f(a), g(b))$. In this case the position p mentioned in the definition is ϵ for the given counterexample. Finally, no TRS including a rule whose left-hand side is $f(f(x))$ is orthogonal, a counterexample is $t = f(f(f(a)))$. In this case the same rule corresponds to l_1 and l_2.

Properties of first-order infinitary orthogonal TRSs are studied e.g. in [KKSdV95, KdV05, EGH$^+$10]. In this thesis, some auxiliary iTRSs we use to define notions pertaining to proof terms, which are "companions" to the *object* iTRS, i.e. the iTRS whose reductions are to be modeled by means of proof terms, happen to be orthogonal. We therefore profit from some properties of orthogonal iTRSs; cfr. Section 5.1.6. This observation does not imply the scope of proof terms to be limited to orthogonal rewriting systems; reductions in any left-linear system, either orthogonal or not, can be described by proof terms. The mentioned "companion" iTRSs are orthogonal, even for non-orthogonal object iTRSs.

Reduction steps

We formalise the notion of *reduction step* as follows.

Definition 5.1.31 (Reduction step, source, target, redex position, depth)**.** *Let* $T = \langle \Sigma, R \rangle$ *be a TRS,* $t \in Ter^\infty(\Sigma)$*,* $p \in \mathtt{Pos}(t)$*,* $\mu : l \to r \in R$ *and* σ *a substitution, such that* $t|_p = \sigma l$*. Then the 4-tuple* $a = \langle t, p, \mu, \sigma \rangle$ *is a* reduction step*. We use* a, a_1, a', b, c*, etc., to denote reduction steps. We define* $src(a) := t$*,* $tgt(a) := t[\sigma r]_p$*,* $\mathtt{rpos}(a) := p$*, and* $d(a) := |p|$*. They are, respectively, the* source*,* target*,* redex position *and* depth *of* a*. We say that* a *is a step* from $src(a)$ to $tgt(a)$*; we use the notation* $t \xrightarrow{a} u$ *to indicate that* a *is a step from* t *to* u*.*

If the source term of a reduction step is clear from the context, it can be omitted when describing the step. On the other hand, if the substitution is unimportant w.r.t. the subject being discussed, it can be omitted as well. Therefore, we will sometimes refer to a reduction step $\langle t, p, \mu, \sigma \rangle$ as $\langle p, \mu, \sigma \rangle$, or even just $\langle p, \mu \rangle$.

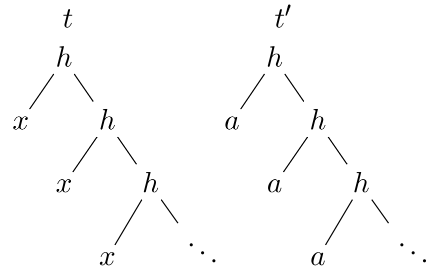
Notice that, given a term t, the reduction steps having t as source term are in an obvious bijection with the occurrences of redexes (i.e. of subterms having the form σl for some rule $\mu : l \to r$) inside t. Namely, the reduction step $\langle t, p, \mu, \sigma \rangle$ correspond to the occurrence, at position p, of a redex with rule μ and substitution σ. Therefore, we take the convention of considering reduction steps from t and **redex occurrences in** t as synonyms. Cfr. Section 1.1.2.

We also want to remark that the definition of a reduction step is given in terms of the *position* of the corresponding redex occurrence, opposed to the *context* which surrounds it (cfr. [BKdV03] dfn. 2.2.4). This decision is motivated by the fact that in infinitary rewriting reasonings, induction on terms (and therefore in contexts which are terms for an extended signature) is not valid, whereas induction on positions is allowed.

Finally, notice that if t, p and μ are known in advance, then the specification of σ is redundant. Nonetheless, we prefer to include the substitution in the definition because it will permit to describe with precision a redex occurrence whose existence is asserted. Notice also that the inclusion of the rule is redundant for orthogonal TRSs; it is included in the characterisation of reduction steps because proof terms are intended to describe reductions in any, maybe non-orthogonal, left-linear TRS.

Some examples of reduction steps follow: consider the TRS whose rules are μ : $f(x) \to g(x)$ and $\nu : h(i(x), y) \to j(y, x)$, and the term $t = g\left(h(i(f(a)), f(i(b)))\right)$. Then there are three reduction steps from t, namely:
$\langle t, 1, \nu, \{x := f(a), y := f(i(b))\}\rangle$, $\langle t, 111, \mu, \{x := a\}\rangle$, and $\langle t, 12, \mu, \{x := i(b)\}\rangle$.

We remark that Dfn. 5.1.30 does not preclude rules having an infinite right-hand side. Consider the term t shown to the right, which can be defined by the recursive equation $t = h(x, t)$. Then $f(x) \to t$ is a valid rewriting rule, allowing the step $f(a) \to t'$, where t', shown also to the right, is defined by the equation $t' = h(a, t')$.



Reduction sequences

In the following, we give a formal definition for the concept of reduction sequence. A precise definition is needed in order to formally establish the relation between stepwise proof terms and reduction sequences, cfr. Section 5.4.2. The formal definition of an infinitary reduction sequence is an essential element of infinitary rewriting; it is discussed in e.g. [KKSdV90, DKP91, KKSdV95, BKdV03, KdV05, Kah10].

A *reduction sequence* is a sequence of reduction steps, having any (finite or infinite) ordinal as length. However, not all sequences of steps are reduction sequences; some conditions must hold. Obviously, if a and b are consecutive steps in a sequence, then $tgt(a)$ must coincide with $src(b)$. Cfr. the corresponding definition given in the ARS model, Dfn. 2.1.10. For infinite sequences, the coherence condition must hold also for steps at *limit* positions. E.g. in a sequence $a_0; a_1; \ldots; a_n; \ldots a_\omega; a_{\omega+1}; \ldots$, there must be some relation between the step a_ω and the sequence of the steps previous to it. This relation is commonly formalised in the literature by asking the sequence of targets of the previous steps, i.e. the sequence $tgt(a_0); tgt(a_1); \ldots; tgt(a_n); \ldots$ to have a limit, and that limit to coincide with $src(a_\omega)$. This requirement is related with the characterisation of *weakly convergent* infinitary rewriting, which is the favored criterion in [DKP91].

We give some examples, using the rules $f(x) \to g(x)$, $g(x) \to k(x)$ and $i(x) \to i(x)$. The successive transformation of each occurrence of f in the term f^ω to g by means of the rule $f(x) \to g(x)$, proceeding outside-in, configure the following, weakly convergent reduction sequence:

$$f^\omega \to g(f^\omega) \to g(g(f^\omega)) \to \ldots \to g(g(\ldots g(f^\omega)\ldots)) \to g(g(\ldots g(g(f^\omega))\ldots)) \to \ldots \tag{5.3}$$

It is not difficult to see that the sequence of the targets in this reduction is convergent, and that its limit is g^ω. Moreover, a subsequent step can be appended to the given reduction sequence, in order to form a new sequence having length $\omega + 1$, iff the source of that step is g^ω. E.g., the concatenation of the sequence (5.3) with the step $g^\omega \to k(g^\omega)$ form a weakly convergent reduction sequence. In turn, if we concatenate (5.3) with an analogously conceived outside-in transformation of each occurrence of g in g^ω to k, in this case by means of the rule $g(x) \to k(x)$, then we obtain the following weakly convergent reduction sequence

$$f^\omega \to g(f^\omega) \to g(g(f^\omega)) \to \dots \; g^\omega \to k(g^\omega) \to k(k(g^\omega)) \to \dots \; k^\omega$$

whose length is $\omega \times 2$. We remark that the term g^ω is not the target of a step in this sequence, but rather the limit of the targets of the first ω steps; hence the absence of an arrow pointing to it. The situation is analogous for k^ω, which is the target of the whole sequence. On a different front, the rule $i(x) \to i(x)$ yields the following, weakly convergent reduction sequence

$$i(a) \to i(a) \to i(a) \to \dots \to i(a) \to i(a) \to \dots$$

whose length is ω, and whose target is $i(a)$.

In order to obtain a notion of reduction sequence enjoying some desired properties, a further condition is imposed. Namely, the *depth* of successive steps is required to tend to infinity at each limit in the sequence, i.e. up to the ω-th step, up to the $\omega * 2$-th step, and so on, and also up to ω^2, etc.. Reduction sequences satisfying this requirement, and also the coherence requirements described before, are known as *strongly convergent* in the literature.[4] This is the criterion favored in e.g. [KKSdV95, KdV05, Ket12]. We adopt the strong conve rgence criterion in this thesis as well. Therefore, in the following definition of reduction sequence, we refer to just *convergence* of reduction sequences, with the meaning of strong convergence.

We observe that the reduction sequence $i(a) \to i(a) \to i(a) \to \dots \to i(a) \to i(a) \to \dots$ described above, is not strongly convergent, because all of its steps are at depth 0. On the other hand, the reduction sequence $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \to \dots \; g^\omega \to k(g^\omega) \to k(k(g^\omega)) \to \dots \; k^\omega$ is strongly convergent: the sequence of the depths of its first ω steps, and also the sequence of the depths of all its steps, tend to infinity.

These considerations motivate the following definitions.

Definition 5.1.32 (Reduction sequence, convergence)**.** *A (well-formed)* reduction sequence *is: either* Id_t*, the* empty reduction sequence for *the term t, or else a non-empty sequence of reduction steps* $\delta := \langle \delta[\alpha] \rangle_{\alpha < \beta}$*, where* $\beta > 0$ *and* δ *verifies all the following conditions:*

1. *For all* α *such that* $\alpha + 1 < \beta$*,* $src(\delta[\alpha + 1]) = tgt(\delta[\alpha])$*.*
2. *For all limit ordinals* $\beta_0 < \beta$*:*
   (a) *The sequence* $\langle tgt(\delta[\alpha]) \rangle_{\alpha < \beta_0}$ *has a limit.*
   (b) $\lim_{\alpha \to \beta_0} tgt(\delta[\alpha]) = src(\delta[\beta_0])$*.*
   (c) *For all* $n < \omega$*, there exists* $\beta' < \beta_0$ *such that* $d(\delta[\alpha]) > n$ *if* $\beta' < \alpha < \beta_0$*.*

[4] In [Kah10], different criteria to formalise the notion of infinitary reduction sequence, including those of weak and strong convergence, are discussed. The notion of *adherence* is proposed there as an alternative to convergence.

We say that a reduction sequence δ is convergent *iff either $\delta = \mathsf{Id}_t$ for some term t, or else $\delta = \langle \delta[\alpha] \rangle_{\alpha<\beta}$, and either β is a successor ordinal, or else β is a limit ordinal and conditions (2a) and (2c) hold for β as well. We use $\delta, \delta_1, \delta', \gamma, \pi$, etc., to denote reduction sequences. We use the semicolon to concatenate reduction sequences.*

Dfn. 5.1.32 coincides with the definitions of reduction sequence and convergent reduction sequence given in [KdV05]. Our definition of reduction sequence coincides with the notion of *strongly contiguous* sequence of steps given in [KKSdV95, BKdV03]. In turn, our definition of strongly convergent reduction sequence is the same as in [BKdV03]; in [KKSdV95], a strongly convergent reduction sequence is a strongly continuous sequence whose length is a successor ordinal.

Definition 5.1.33 (Source of a reduction sequence)**.** *Let δ be a reduction sequence. We define the* source *term of δ, notation $src(\delta)$, as follows: if $\delta = \mathsf{Id}_t$, then $src(\delta) := t$, if $\delta = \langle \delta[\alpha] \rangle_{\alpha<\beta}$, then $src(\delta) := src(\delta[0])$.*

Definition 5.1.34 (Target of a reduction sequence)**.** *Let δ be a convergent reduction sequence. We define the the* target *term of δ, notation $tgt(\delta)$, as follows: if $\delta = \mathsf{Id}_t$, then $tgt(\delta) := t$; if $\delta = \langle \delta[\alpha] \rangle_{\alpha<\beta}$, then $\beta = \beta' + 1$ implies $tgt(\delta) := tgt(\delta[\beta'])$, and β being a limit ordinal implies $tgt(\delta) := \lim_{\alpha \to \beta} tgt(\delta[\alpha])$.*

Definition 5.1.35 (Length of a reduction sequence)**.** *Let δ be a reduction sequence. We define the* length *of δ, notation $\mathsf{length}(\delta)$, as follows: if $\delta = \mathsf{Id}_t$, then $\mathsf{length}(\delta) := 0$, if $\delta = \langle \delta[\alpha] \rangle_{\alpha<\beta}$, then $\mathsf{length}(\delta) := \beta$.*

Remark 5.1.36. *Any mention of $tgt(\delta)$ implies that the target of the reduction sequence δ is defined, i.e. that δ is a* convergent *reduction sequence.*

Notation 5.1.37. *We write $t \xrightarrow{\delta}\!\!\!\twoheadrightarrow u$ to denote that δ is a convergent reduction sequence, $src(\delta) = t$ and $tgt(\delta) = u$. If $\mathsf{length}(\delta) < \omega$, then we write $t \xrightarrow{\delta}\!\!\twoheadrightarrow u$ as well. We also use the notation $t \twoheadrightarrow\!\!\!\twoheadrightarrow u$ ($t \twoheadrightarrow u$), to denote that t can be transformed into u by means of a (finite) reduction sequence.*

Definition 5.1.38 (Minimum activity depth of a reduction sequence)**.** *Let δ be a reduction sequence. We define the* minimum activity depth *of δ, notation $mind(\delta)$, as follows: if $\delta = \mathsf{Id}_t$, then $mind(\delta) := \omega$, if $\delta = \langle \delta[\alpha] \rangle_{\alpha<\beta}$, then $mind(\delta) := min\{d(\delta[\alpha]) \mid \alpha < \beta\}$.*

Definition 5.1.39 (Fragment of a reduction sequence)**.** *Let δ be a reduction sequence and α, β ordinals verifying $\alpha < \mathsf{length}(\delta)$, $\beta \leqslant \mathsf{length}(\delta)$ and $\alpha \leqslant \beta$. We define the* fragment *of δ from α to β, notation $\delta[\alpha, \beta)$, as follows: if $\alpha = \beta < \mathsf{length}(\delta)$, then $\delta[\alpha, \beta) := \mathsf{Id}_{src(\delta[\alpha])}$, otherwise, i.e. if $\alpha < \beta$, then $\delta[\alpha, \beta) := \langle \delta[\alpha + \gamma] \rangle_{\gamma \,/\, \alpha+\gamma<\beta}$.*

We give some examples of infinite reduction sequences, using the following rules: $\mu : f(x) \to g(x)$, $\nu : g(x) \to k(x)$, $\tau : a \to k(a)$, $\rho : i(x) \to i(x)$, $\theta : b \to j(b, b)$, $\eta : k(x) \to c$.

A simple example of convergent reduction sequence is

$$f^\omega \to g(f^\omega) \to g(g(f^\omega)) \to \ldots \to g(g(\ldots g(f^\omega)\ldots)) \twoheadrightarrow\!\!\!\twoheadrightarrow g^\omega$$

The length of this reduction sequence is ω; its n-th step is $\langle g^n(f^\omega), 1^n, \mu, \{x := f^\omega\}\rangle$. For any countable ordinal λ, it is easy to define a convergent reduction sequence $f^\omega \xrightarrow{\delta_\lambda}\!\!\!\twoheadrightarrow g^\omega$

such that $\mathsf{length}(\delta_\lambda) = \lambda$: it suffices to consider a bijective function $F : \lambda \to \omega$, and to define, for all $\alpha < \lambda$, $\delta_\lambda[\alpha]$ as the transformation of the occurrence of f at depth $F(\alpha)$ into an occurrence of g, by means of the μ rule. E.g., if $\lambda = \omega * 2$, $F(n) = 2n + 1$ if $n < \omega$, and $F(\omega + n) = 2n$, then we have the following reduction sequence:

$$f^\omega \to f(g(f^\omega)) \to f(g(f(g(f^\omega)))) \twoheadrightarrow t' \to g(g(t')) \to g(g(g(g(t')))) \twoheadrightarrow g^\omega$$

where t' is the term defined by the equation $t' = f(g(t'))$. Another convergent reduction sequence whose length is $\omega * 2$ is

$$f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega \to k(g^\omega) \to k(k(g^\omega)) \twoheadrightarrow k^\omega$$

Notice that this reduction sequence is equivalent, in the sense that involves the same steps, as the following

$$f^\omega \to g(f^\omega) \to k(f^\omega) \to k(g(f^\omega)) \to k(k(f^\omega)) \twoheadrightarrow k^\omega$$

so that equivalent infinite reductions can have different lengths, even when no erasure is involved.

The reduction sequence

$$i(a) \to i(a) \to i(a) \to \dots$$

whose length is ω, is not convergent because the depth of all its steps is 0: all the steps in this sequence have the form $\langle i(a), \epsilon, \rho, \{x := a\}\rangle$. On the other hand, infinite convergent reduction sequences exist for which both their source and target are finite terms. An example follows.

$$a \to k(a) \to k(k(a)) \twoheadrightarrow k^\omega \to c$$

The depth condition of strong convergence, along with the fact that the left-hand side of any rewrite rule must be a finite term, imply that the target of any convergent reduction sequence whose length is a *limit* ordinal, must be an infinite term.

Finally, we note that an infinite reduction sequence can involve rewrite rules in which more than one variable appear, as in the following example:

$$b \to j(b, b) \to j(j(b, b), b) \twoheadrightarrow u$$

where u is the term defined by the equation $u = j(u, b)$.

It is worth remarking that the requirement about depths of successive steps, i.e. condition (2c) in Dfn. 5.1.32, is not enough to guarantee the well-formedness of reduction sequences. Let us discuss briefly this issue. Some examples will be given using the rules $f(x) \to g(x)$, $h(x) \to j(x)$, and $g(x) \to f(x)$.

The depth requirement alone does not guarantee coherence at limit positions, as discussed prior to Dfn. 5.1.32. E.g., the sequence of steps which results of the concatenation of $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega$ and $h^\omega \to j(h^\omega) \to j(j(h^\omega)) \twoheadrightarrow j^\omega$, which total length is $\omega * 2$, does not produce a well-formed reduction sequence, even when depths tend to infinity at each limit ordinal in the sequence of steps; a target (namely g^ω) can be determined for the prefix of the first ω steps, but it does not coincide with the source of the ω-th step, i.e. h^ω.

Moreover, the depth condition alone does not even guarantee the existence of a limit for each limit ordinal prefix. E.g. consider the sequence of steps, having length ω^2, informally described as follows: $f^\omega \twoheadrightarrow g^\omega; g^\omega \twoheadrightarrow f^\omega; g(f^\omega) \twoheadrightarrow g^\omega; f(g^\omega) \twoheadrightarrow f^\omega; g^2(f^\omega) \twoheadrightarrow g^\omega; f^2(g^\omega) \twoheadrightarrow f^\omega; \ldots g^n(f^\omega) \twoheadrightarrow g^\omega; f^n(g^\omega) \twoheadrightarrow f^\omega; \ldots$, where each fragment includes ω steps performed from the outside in, and the semicolon denotes concatenation of sequences. This sequence of steps obeys the depth condition at each limit ordinal, including ω^2 itself, but however, a limit cannot be determined for it. Therefore, the requirement about the existence of a limit, i.e. condition (2a), cannot be removed by the mere fact of including the depth requirement.

It could possibly be proved, by means of a careful transfinite induction on limit ordinals, that for any sequence of steps, and each limit ordinal β up to the length of that sequence, the depth requirement on each limit ordinal $\leqslant \beta$, plus coherence (i.e. condition (2b)) at all limit ordinals $< \beta$, imply the existence of a limit in the sequence of targets at ordinal β. Since this issue is not in the focus of the present work, we leave it as subject of further investigation.

Other notions

The definition of reduction step leads immediately to that of *normal form*; cfr. Dfn. 2.1.8.

Definition 5.1.40 (Normal form). *A* ***normal form*** *is a term having no redex occurrences, or equivalently, a term being the source of no reduction step.*

Given a term t, we will refer to the reduction sequences having t as source term as the reduction sequences **from** t. Moreover, if s is the only normal form verifying $t \twoheadrightarrow s$, then we will say that s is the (infinitary) **normal form of** t.

We can define reduction steps and sequences which model applications of rules to contexts rather than terms.

Remark 5.1.41. *For any TRS $T = \langle \Sigma, R \rangle$ we can think of an associated TRS $T^\Box := \langle \Sigma \cup \{\Box/0\}, R \rangle$, which makes it possible to describe reductions on contexts. In the sequel we will include references to reduction steps and reduction sequences whose source and target are contexts; they must be understood as defined in $T^\Box$.*

5.1.5 Patterns, pattern depth

Given a rewrite rule $\mu : l \to r$ and a reduction step $a = \langle t, p, \mu, \sigma \rangle$, the role of the *function symbol* occurrences in l differs from that of the *variable* occurrences: the former must be present explicitly in $src(a)$ having the same structure as in l; while the latter are included in the domain of σ.

We will sometimes need to refer to the positions of all the occurrences of function symbols in (the lhs of) a rule, and also in (the source term of) a reduction step. E.g. if $\mu = f(g(x, h(y))) \to f(y)$, then the occurrences of function symbols in (the lhs of) μ are at positions ϵ, 1 and 12. The corresponding formal definitions follow.

Definition 5.1.42 (Pattern, pattern positions, pattern depth). *Let t be a term. The* pattern *of t, notation $\mathtt{pat}(t)$, is the context which results of changing all the variable occurrences in t with boxes; cfr. [BKdV03] dfn. 2.7.3, pg. 49. The set of* pattern positions *of t, notation $\mathtt{Ppos}(t)$ is defined as $\{p \ / \ p \in \mathtt{Pos}(t) \text{ and } t(p) \notin \mathtt{Var}\}$. The* pattern depth *of t, notation $\mathtt{Pd}(t)$, is defined as $max(\{|p| \ / \ p \in \mathtt{Ppos}(t)\})$; if $x \in \mathtt{Var}$ then $\mathtt{Pd}(x)$ is undefined.*

Let $\mu : l \to r$ be a rewrite rule. The set of pattern positions and the pattern depth of μ are defined as follows: $\mathtt{Ppos}(\mu) := \mathtt{Ppos}(l)$*,* $\mathtt{Pd}(\mu) := \mathtt{Pd}(l)$*.*

Let $a = \langle t, p, \mu, \sigma \rangle$ be a reduction step. The set of pattern positions of a is defined as follows: $\mathtt{Ppos}(a) := p \cdot \mathtt{Ppos}(\mu)$*.*

For example, if $\mu : h(i(x), g(i(y)), c) \to h(x, x, y)$, $t = g(h(i(g(a)), g(i(b)), c))$ and $a = \langle t, 1, \mu, \{x := g(a), y := b\} \rangle$, then $\mathtt{Ppos}(\mu) = \{\epsilon, 1, 2, 21, 3\}$, $\mathtt{Pd}(\mu) = 2$, and $\mathtt{Ppos}(a) = 1 \cdot \mathtt{Ppos}(\mu) = \{1, 11, 12, 121, 13\}$.

5.1.6 Some properties about infinitary rewriting

We include in this section the statement and proof of some properties on infinitary rewriting which are needed in following sections. In turn, these properties require some definitions to be given.

We say that a term t is *infinitary weakly normalising*, shorthand notation WN^∞, iff there exists at least one reduction sequence δ such that $t \xrightarrow{\delta}\!\!\!\twoheadrightarrow u$ and u is a normal form. We say that a term t is *strongly normalising*, shorthand notation SN^∞, iff there is no divergent reduction sequence whose source term is t. A term t has the *unique normal-form property*, shorthand notation UN^∞, iff whenever $t \twoheadrightarrow u_1$, $t \twoheadrightarrow u_2$ and both u_1 and u_2 are normal forms, then $u_1 = u_2$. A TRS is WN^∞ (SN^∞, UN^∞) iff all its terms are. Cfr. [KdV05] for a study of normalisation for infinitary rewriting.

A TRS T is *disjoint* iff the set of all the function symbols occurring in the left-hand sides of the rules of T is disjoint from the set of all the function symbols occurring in the right-hand sides of the rules of T.

The results to be given in this section are particularly needed for the study of the class of proof terms corresponding to coinitial sets of redexes, which involves the definition of TRSs which are 'companions' to the TRS under study. Cfr. the concept of *2-rewriting system*, notation 8.2.12 in [BKdV03].

The 'companion' TRSs enjoy some desirable properties. First of all, they are all orthogonal, and therefore they enjoy the property UN^∞; cfr. [KdV05] Section 5. Some of them are Recursive Program Schemes (cfr. [BKdV03] dfn. 3.4.7), i.e., they are orthogonal and all their rules have the form $f(\ldots, x_i, \ldots) \to t$, so that we can distinguish the subset $\mathcal{F} := \{f \ / \ f(\ldots, x_i, \ldots) \to t \in R\}$ within their signature.

Notice that for Recursive Program Schemes, the disjointness condition amounts to assert that no symbol in $\mathcal{F}$ appears in the right-hand side of any rule.

Fragments of reduction sequences, cfr. Dfn. 5.1.39, enjoy some basic properties.

Lemma 5.1.43. *Let δ be a reduction sequence, and $\alpha <$ $\mathsf{length}(\delta)$. Then $\delta[0, \alpha)$ is convergent.*

Proof. It is immediate to verify that $\delta[0, \alpha)$ is a well-formed reduction sequence. If $\alpha = 0$, i.e. $\delta[0, \alpha) = \mathsf{Id}_{src(\delta)}$, or if α is a successor ordinal, then it is immediately convergent. If α is a limit ordinal, the fact that δ is well-formed implies that conditions (2a) and (2c) hold for $\alpha < \mathsf{length}(\delta)$, hence $\delta[0, \alpha)$ is convergent. □

Lemma 5.1.44. *Let δ be a reduction sequence and $\alpha <$ $\mathsf{length}(\delta)$. Then $src(\delta[\alpha]) = tgt(\delta[0, \alpha))$.*

Proof. Notice that Lem. 5.1.43 implies that $\delta[0,\alpha)$ is convergent, so that its limit is defined. If $\alpha = 0$, i.e. $\delta[0,\alpha) = \mathsf{Id}_{src(\delta[0])}$, then we conclude immediately. Otherwise, $\alpha = \alpha' + 1$ implies $src(\delta[\alpha]) = tgt(\delta[\alpha'])$, and α limit implies $src(\delta[\alpha]) = \lim_{\alpha' \to \alpha} tgt(\delta[\alpha'])$, cfr. conditions (1) and (2b) resp. in Dfn. 5.1.32. In either case, this coincides with $tgt(\delta[0,\alpha))$, cfr. Dfn. 5.1.34. Thus we conclude. □

We prove some expected properties of targets of convergent reduction sequences.

Lemma 5.1.45. *Let δ be a convergent reduction sequence and $n < \omega$ such that $mind(\delta) > n$. Then $\mathtt{dist}(src(\delta), tgt(\delta)) < 2^{-n}$.*

Proof. We proceed by induction on $\mathsf{length}(\delta)$. If $\mathsf{length}(\delta) = 0$, i.e. $\delta = \mathsf{Id}_t$ for some term t, then $tgt(\delta) = src(\delta) = t$, so that we conclude immediately.

Assume that $\mathsf{length}(\delta)$ is a successor ordinal, so that $\delta = \delta'; a$ where $\mathsf{length}(\delta') < \mathsf{length}(\delta)$. Then IH can be applied to obtain $\mathtt{dist}(src(\delta'), tgt(\delta')) = \mathtt{dist}(src(\delta), src(a)) < 2^{-n}$. In turn, $tgt(\delta) = tgt(a) = src(a)[s]_p$ for some term s, where $p = \mathtt{rpos}(a)$, so that hypotheses imply $mind(a) > n$. Then Lem. 5.1.26 implies $\mathtt{dist}(src(a), tgt(\delta)) \leqslant 2^{-|p|} < 2^{-n}$. Hence Lem. 5.1.25 allows to conclude.

Assume that $\alpha := \mathsf{length}(\delta)$ is a limit ordinal. In this case $tgt(\delta) = \lim_{\alpha' \to \alpha} tgt(\delta[\alpha'])$. Let $\alpha_n < \alpha$ such that $\mathtt{dist}(tgt(\delta[\alpha']), tgt(\delta)) < 2^{-n}$ if $\alpha_n < \alpha' < \alpha$. Then particularly $\mathtt{dist}(tgt(\delta[\alpha_n + 1]), tgt(\delta)) = \mathtt{dist}(tgt(\delta[0, \alpha_n + 2)), tgt(\delta)) < 2^{-n}$; recall $\alpha_n < \alpha$ limit implies $\alpha_n + 2 < \alpha$. In turn, IH can be applied on $\delta[0, \alpha_n + 2)$ to obtain $\mathtt{dist}(src(\delta[0, \alpha_n + 2)), tgt(\delta[0, \alpha_n + 2))) < 2^{-n}$. Hence we conclude by Lem. 5.1.25. □

Lemma 5.1.46. *Let $t \xrightarrow{\delta}\!\!\!\!\twoheadrightarrow u$ and $p \in \mathtt{Pos}(t)$ such that $\mathtt{rpos}(\delta[\alpha]) \parallel p$ for all $\alpha <$ $\mathsf{length}(\delta)$. Then $t|_p = u|_p$.*

Proof. We proceed by induction on $\mathsf{length}(\delta)$. If $\mathsf{length}(\delta) = 0$, i.e. $\delta = \mathsf{Id}_t$, then we conclude immediately since $u = t$.

Assume that $\mathsf{length}(\delta)$ is a successor ordinal, so that $t \xrightarrow{\delta'}\!\!\!\!\twoheadrightarrow u' \xrightarrow{a} u$. In this case, IH applies to δ', yielding $t|_p = u'|_p$. In turn, $u = u'[s]_q$ for some term s, where $q = \mathtt{rpos}(a) \parallel p$. Then Lem. 5.1.17 implies $u'|_p = u|_p$. Thus we conclude.

Assume that $\alpha := \mathsf{length}(\delta)$ is a limit ordinal. In this case we have $u = tgt(\delta) = \lim_{\alpha' \to \alpha} tgt(\delta[\alpha'])$. Let $n < \omega$, and $\alpha_n < \alpha$ such that $\mathtt{dist}(tgt(\delta[\alpha']), u) < 2^{-(n+|p|)}$, implying $\mathtt{dist}(tgt(\delta[\alpha'])|_p, u|_p) < 2^{-n}$, if $\alpha_n < \alpha' < \alpha$. Recall that $\alpha_n < \alpha$ limit implies $\alpha_n + i < \alpha$ if $i < \omega$. Then $\mathtt{dist}(tgt(\delta[\alpha_n + 1])|_p, u|_p) < 2^{-n}$. Moreover, IH can be applied to $\delta[0, \alpha_n + 2)$, yielding $src(\delta[0, \alpha_n + 2))|_p = tgt(\delta[0, \alpha_n + 2))|_p$, so that $t|_p = tgt(\delta[\alpha_n + 1])|_p$. Hence $\mathtt{dist}(t|_p, u|_p) < 2^{-n}$ for all $n < \omega$. Consequently, we conclude. □

The properties just introduced allow to define the *projection* of a reduction sequence not including head steps over an index. We verify that the definition yields a well-formed reduction sequence; in the infinitary setting, this verification involves a fair amount of work. The following definition involves the use of a sequence of non-contiguous ordinals which we will call A. We use $\mathsf{ord}(A)$ and $A[\alpha]$ to denote the order type of A and its α-th element respectively, where $\alpha < \mathsf{ord}(A)$. In turn, this sequence is built from a set of ordinals S as follows. If $S = \varnothing$, then A is the empty sequence, so that $\mathsf{ord}(A) = 0$. Otherwise, we define $A[0]$ as the minimal element of S. Let $\alpha > 0$ such that $A[\alpha']$ is defined for all $\alpha' < \alpha$. If $\alpha = \alpha' + 1$ then we consider the set $\{\beta \in S \ / \ \beta > A[\alpha']\}$, and if α is a limit ordinal then we consider $\{\beta \in S \;/\; \beta \geqslant sup(\{A[\alpha'] \;/\; \alpha' < \alpha\})\}$. In either case, if the considered set is empty then we state that $A[\alpha_1]$ as undefined for all $\alpha_1 \geqslant \alpha$, so that $\mathsf{ord}(A) = \alpha$. Otherwise, we define $A[\alpha]$ as the minimum of the considered set.

Definition 5.1.47. *Let δ be a reduction sequence such that $mind(\delta) > 0$, and i such that $1 \leqslant i \leqslant m$ where $src(\delta) = f(t_1, \ldots, t_m)$. We define the* projection *of δ* over *i, notation $\delta\,|_i$, as the reduction sequence whose specification follows.*

Let A be the sequence built from the set $\{\alpha \;/\; \alpha < \mathsf{length}(\delta) \wedge i \leqslant \mathtt{rpos}(\delta[\alpha])\}$, w.r.t. the usual order of ordinals. If A is empty, then $\delta\,|_i := \mathsf{Id}_{t_i}$. Otherwise $\mathsf{length}(\delta\,|_i) :=$ $\mathsf{ord}(A)$, and $(\delta\,|_i)[\alpha] := \langle s_i, p, \mu\rangle$ where $\delta[A[\alpha]] = \langle f(s_1, \ldots, s_i, \ldots, s_m), ip, \mu\rangle$. Observe that Lem. 5.1.43 implies $\delta[0, A[\alpha])$ to be convergent, and in turn Lem. 5.1.45 implies $tgt(\delta[0, A[\alpha]))(\epsilon) = src(\delta)(\epsilon) = f$; therefore, $tgt(\delta[0, A[\alpha])) = src(\delta[A[\alpha]]) =$ $f(s_1, \ldots, s_i, \ldots, s_m)$. Cfr. also Lem. 5.1.44.

Lemma 5.1.48. *Let δ be a reduction sequence such that $mind(\delta) > 0$, and i such that $1 \leqslant i \leqslant m$ where $src(\delta)(\epsilon) = f/m$. Then $\delta\,|_i$ is a well-formed reduction sequence and $src(\delta\,|_i) = src(\delta)\,|_i$. Moreover, if δ is convergent, then $\delta\,|_i$ is convergent as well, and $tgt(\delta\,|_i) = tgt(\delta)|_i$.*

Proof. Let A be the sequence of positions of steps in δ at or below position i. We proceed by induction on $\mathsf{length}(\delta\,|_i) = \mathsf{ord}(A)$.

Assume A is empty, so that $\delta\,|_i = \mathsf{Id}_{src(\delta)_i}$. Then just Dfn. 5.1.32 implies immediately that $\delta\,|_i$ is a well-formed and convergent reduction sequence, and Dfn. 5.1.33 that $src(\delta\,|_i) = src(\delta)\,|_i$. If δ is convergent, then observe that A being empty implies $\mathtt{rpos}(\delta[\alpha]) \parallel i$ for all $\alpha < \mathsf{length}(\delta)$; recall $mind(\delta) > 0$. Then Lem 5.1.46 implies $tgt(\delta)|_i = src(\delta)|_i = tgt(\delta\,|_i)$. Thus we conclude.

Assume that $\mathsf{ord}(A) = \alpha + 1$, i.e., $\mathsf{ord}(A)$ is a successor ordinal. Observe that $(\delta\,|_i)[0, \alpha) = \delta[0, A[\alpha])\,|_i$, and that Lem. 5.1.43 implies that $\delta[0, A[\alpha])$ is convergent. Then IH on $\delta[0, A[\alpha])$ yields that $(\delta\,|_i)[0, \alpha)$ is a well-formed and convergent reduction sequence, that $src(\delta\,|_i) = src((\delta\,|_i)[0, \alpha)) = src(\delta)\,|_i$, and that $tgt((\delta\,|_i)[0, \alpha)) =$ $tgt(\delta[0, A[\alpha]))\,|_i = src(\delta[A[\alpha]])\,|_i$, cfr. Lem. 5.1.44. On the other hand, $src((\delta\,|_i)[\alpha]) =$ $src(\delta[A[\alpha]])\,|_i$.

We verify that the conditions in Dfn. 5.1.32 hold for $\delta\,|_i$. The analysis depends on α.

- If $\alpha = 0$, then $\delta\,|_i[0, \alpha) = \mathsf{Id}_{src(\delta)_i}$. In this case, conditions (1) and (2) hold immediately.
- If $\alpha = \alpha' + 1$, then $(\delta\,|_i)[0, \alpha)$ being a well-formed reduction sequence implies that condition (1) holds for all α_0 such that $\alpha_0 + 1 < \alpha$; i.e. for all needed indexes but α'. In turn, $tgt((\delta\,|_i)[\alpha']) = tgt((\delta\,|_i)[0, \alpha)) = src(\delta[A[\alpha]])\,|_i =$ $src((\delta\,|_i)[\alpha]) = src((\delta\,|_i)[\alpha' + 1])$. On the other hand, $(\delta\,|_i)[0, \alpha)$ being well-formed implies also that condition (2) holds for $\delta\,|_i$; indeed, $\alpha_0 < (\alpha' + 1) + 1$ and α_0 limit implies $\alpha_0 < \alpha' + 1$.
- If α is a limit ordinal, then $(\delta\,|_i)[0, \alpha)$ being a well-formed reduction sequence implies that condition (1) holds for $\delta\,|_i$; notice $\alpha_0 + 1 < \alpha + 1$ implies $\alpha_0 < \alpha$, so that α limit implies in turn $\alpha_0 + 1 < \alpha$. Furthermore, $(\delta\,|_i)[0, \alpha)$ being convergent implies that conditions (2a) and (2c) hold for all α_0 limit ordinals verifying $\alpha_0 < \alpha+1$, particularly for α; and also that condition (2b) holds for all limit $\alpha_0 < \alpha$. In turn, $\lim_{\alpha' \to \alpha} tgt((\delta\,|_i)[\alpha']) = tgt((\delta\,|_i)[0, \alpha)) = src(\delta[A[\alpha]])|_i$ $= src((\delta\,|_i)[\alpha])$, so that condition (2b) to hold also for α

Hence, in either case, we have verified that $\delta|_i$ is a well-formed reduction sequence. In turn, $\mathsf{length}(\delta|_i) = \mathsf{ord}(A)$ being a successor ordinal implies immediately that $\delta|_i$ is convergent.

If δ is convergent, then we must verify $tgt(\delta|_i) = tgt(\delta)|_i$. Let $(\delta|_i)[\alpha] = \langle t_i, p, \mu\rangle$ where $\delta[A[\alpha]] = \langle f(t_1, \ldots, t_i, \ldots, t_m), ip, \mu\rangle$. Then $tgt(\delta|_i) = tgt((\delta|_i)[\alpha]) = t_i[s]_p$ for some term s, and $tgt(\delta[A[\alpha]]) = f(t_1, \ldots, t_i, \ldots, t_m)[s]_{ip} = f(t_1, \ldots, t_i[s]_p, \ldots, t_m)$, cfr. Lem. 5.1.15, therefore $tgt(\delta|_i) = tgt(\delta[A[\alpha]])|_i$. If $\mathsf{length}(\delta) = A[\alpha] + 1$, then $tgt(\delta) = tgt(\delta[A[\alpha]])$. Otherwise, for all α' verifying $A[\alpha] < \alpha' < \mathsf{length}(\delta)$, it is immediate that $\mathtt{rpos}(\delta[\alpha']) \parallel i$. Then Lem. 5.1.46 implies $tgt(\delta[A[\alpha]+1, \mathsf{length}(\delta)))|_i = src(\delta[A[\alpha]+1, \mathsf{length}(\delta)))|_i$. In either case, $tgt(\delta)|_i = tgt(\delta[A[\alpha]])|_i = tgt(\delta|_i)$. Thus we conclude.

Assume that $\alpha := \mathsf{ord}(A)$ is a limit ordinal.

Let α' such that $\alpha' + 1 < \alpha$, then α limit implies $\alpha' + 2 < \alpha$. Therefore IH can be applied to obtain that $(\delta|_i)[0, \alpha'+2)$ is a well-formed reduction sequence, implying that $src((\delta|_i)[\alpha'+1]) = tgt((\delta|_i)[\alpha'])$. Consequently, $\delta|_i$ verifies condition (1) in Dfn. 5.1.32.

Let α_0 be a limit ordinal verifying $\alpha_0 < \alpha$. Observe that $A[\alpha_0] < \mathsf{length}(\delta)$, then Lem. 5.1.43 implies that $\delta[0, A[\alpha_0])$ is convergent. We apply IH to obtain that $(\delta|_i)[0, \alpha_0)$ is a well-formed and convergent reduction sequence. Therefore conditions (2a) and (2c) hold for $\delta|_i$ w.r.t. α_0. Moreover $\lim_{\alpha' \to \alpha_0} tgt((\delta|_i)[\alpha']) = tgt((\delta|_i)[0, \alpha_0)) = src((\delta|_i)[\alpha_0]$, cfr. Dfn. 5.1.34 and Lem. 5.1.44 resp.. Hence $\delta|_i$ enjoys condition (2b) w.r.t. α_0 as well.

Consequently, $\delta|_i$ is a well-formed reduction sequence. Observe that $src(\delta|_i) = src((\delta|_i)[0]) = src(\delta[0, A[1])|_i)$. Since obviously $1 < \alpha$, we can use IH to obtain $src(\delta|_i) = src(\delta[0, A[1]))|_i = src(\delta)|_i$.

Assume that δ is convergent. Let $B := \{\beta' \,/\, \beta' < \mathsf{length}(\delta) \wedge A[\alpha'] < \beta'$ for all $\alpha' < \alpha\}$. We define β as follows: $\beta := \mathsf{length}(\delta)$ if B is empty, and $\beta := min(B)$ otherwise. Assume for contradiction that $\beta = \beta' + 1$ for some β'. If B is empty, so that $\mathsf{length}(\delta) = \beta' + 1$, then $\beta' \notin B$ implies the existence of some $\alpha' < \alpha$ such that $\beta' \leqslant A[\alpha']$ and then $\beta' < A[\alpha'+1]$, contradicting $A[\alpha'+1] < \mathsf{length}(\delta)$. Otherwise $\beta = min(B)$, implying that $\beta' \leqslant A[\alpha']$ for some $\alpha' < \alpha$. But this would imply $\beta \leqslant A[\alpha'+1]$, contradicting $\beta \in B$. Consequently, β is a limit ordinal.

We verify conditions (2a) and (2c) for $\delta|_i$ w.r.t. α.

- To verify condition (2a), it is enough to show that $\lim_{\alpha' \to \alpha} tgt((\delta|_i)[\alpha']) = u|_i$, where $u = \lim_{\beta' \to \beta} tgt(\delta[\beta']) = tgt(\delta[0, \beta))$. Let $n < \omega$, and $\beta_n < \beta$ such that $\mathtt{dist}(tgt(\delta[\beta']), u) < 2^{-(n+1)}$, implying $\mathtt{dist}(tgt(\delta[\beta'])|_i, u|_i) < 2^{-n}$, if $\beta_n < \beta' < \beta$. Then $\beta_n < \beta$ implies that $\beta_n \leqslant A[\alpha_n]$ for some $\alpha_n < \alpha$, then $\alpha_n < \alpha' < \alpha$ implies $\mathtt{dist}(tgt((\delta|_i)[\alpha']), u|_i) < 2^{-n}$, recalling that $tgt((\delta|_i)[\alpha']) = tgt(\delta[A[\alpha]])|_i$. Consequently, $\lim_{\alpha' \to \alpha} tgt((\delta|_i)[\alpha']) = tgt(\delta[0, \beta))|_i$, and then $\delta|_i$ verifies condition (2a) w.r.t. α.
- Let $n < \omega$, let $\beta_n < \beta$ such that $d(\delta[\beta']) > n + 1$ if $\beta_n < \beta' < \beta$. By an argument similar to that used for condition (2a), we obtain the existence of some $\alpha_n < \alpha$ such that $d(\delta[A[\alpha']]) > n + 1$, implying $d((\delta|_i)[\alpha']) > n$, if $\alpha_n < \alpha' < \alpha$. Consequently, $\delta|_i$ verifies condition (2c) for α.

Hence, $\delta|_i$ is a convergent reduction sequence. In turn, Dfn. 5.1.34 yields $tgt(\delta|_i) = \lim_{\alpha' \to \alpha} tgt((\delta|_i)[\alpha'])$, then we have already verified that $tgt(\delta|_i) = tgt(\delta[0, \beta))|_i$. If $\beta = \mathsf{length}(\delta)$, then immediately $tgt(\delta|_i) = tgt(\delta)|_i$. Otherwise, it is immediate to observe that $\mathtt{rpos}(\delta[\beta']) \parallel i$ if $\beta \leqslant \beta' < \mathsf{length}(\delta)$. Hence $tgt(\delta|_i) = tgt(\delta[0, \beta))|_i$

$= src(\delta[\beta, \mathsf{length}(\delta))) \mid_i = tgt(\delta[\beta, \mathsf{length}(\delta))) \mid_i = tgt(\delta) \mid_i$; by already obtained result, Lem. 5.1.44 (recall $src(\delta[\beta, \mathsf{length}(\delta))) = src(\delta[\beta])$, Lem. 5.1.46, and simple analysis of Dfn. 5.1.34 resp.. Thus we conclude. $\square$

The following result extends the idea of a projection of a reduction sequence from arguments of function symbols to arguments of contexts.

Lemma 5.1.49. *Let C a context having exactly m holes, and $C[t_1, \ldots, t_m] \stackrel{\delta}{\twoheadrightarrow} u$, such that for all $\alpha < \mathit{length}(\delta)$, there exists some i verifying $1 \leqslant i \leqslant m$ and $\mathtt{Bpos}(C, i) \leqslant \mathtt{rpos}(\delta[\alpha])$. Then $u = C[u_1, \ldots, u_m]$ and for all i such that $1 \leqslant i \leqslant m$, there is a reduction sequence δ_i verifying $t_i \stackrel{\delta_i}{\twoheadrightarrow} u_i$.*

Proof. Straightforward induction on $max\{|\mathtt{Bpos}(C, i)| \ / \ 1 \leqslant i \leqslant m\}$, resorting on Lem. 5.1.48 for the inductive case. $\square$

We illustrate Lem. 5.1.49 by means of an example, using the rules $f(x) \to g(x)$ and $k(x) \to j(x)$. Let us consider the sequence δ defined as follows:

$$\begin{array}{rcl} t = h(m(f^\omega), m(k^\omega)) & \to & h(m(g(f^\omega)), m(k^\omega)) \to h(m(g(f^\omega)), m(j(k^\omega))) \\ & \to & h(m(g(g(f^\omega))), m(j(k^\omega))) \to h(m(g(g(f^\omega))), m(j(j(k^\omega)))) \\ & \twoheadrightarrow & h(m(g^\omega), m(j^\omega)) = u \end{array}$$

and the context $C = h(m(\Box), m(\Box))$, so that $\mathtt{Bpos}(C, 1) = 11$, $\mathtt{Bpos}(C, 2) = 21$, and $t = C[t_1, t_2]$ where $t_1 = f^\omega$ and $t_2 = k^\omega$. Notice that $11 \leqslant \mathtt{rpos}(\delta[n])$ if n is odd, and $21 \leqslant \mathtt{rpos}(\delta[n])$ if n is even. Therefore, δ verifies the lemma hypotheses. Observe that $u = C[g^\omega, j^\omega]$, $f^\omega \stackrel{\delta_1}{\twoheadrightarrow} g^\omega$ and $k^\omega \stackrel{\delta_2}{\twoheadrightarrow} j^\omega$, where δ_1 is exactly the sequence obtained by projecting the steps in δ having odd indexes on the position 11, namely $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \to \ldots$, and analogously for δ_2.

Two properties about normalisation follow.

Lemma 5.1.50. *Let T an orthogonal TRS, and t, s, u terms such that $t \stackrel{\pi}{\twoheadrightarrow} u$, $t \stackrel{\delta}{\twoheadrightarrow} s$, u is a normal form, and $d(\delta[i]) = 0$ for all $i < \mathit{length}(\delta)$. Then $s \stackrel{\pi'}{\twoheadrightarrow} u$ for some reduction sequence π'.*

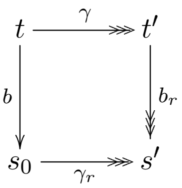
Proof. We proceed by induction on $\mathsf{length}(\delta)$; observe that δ is finite, so that non-transfinite induction suffices. If $\mathsf{length}(\delta) = 0$, i.e. δ is the empty reduction for t, then $s = t$ so that we conclude by taking $\pi' := \pi$.

Assume $\mathsf{length}(\delta) = n + 1$, so that $t \xrightarrow{a} s_0 \stackrel{\delta'}{\twoheadrightarrow} s$ where $a = \langle t, \epsilon, \mu \rangle$ for some rule $\mu : l[x_1, \ldots, x_m] \to h$, and $\mathsf{length}(\delta') = n$.

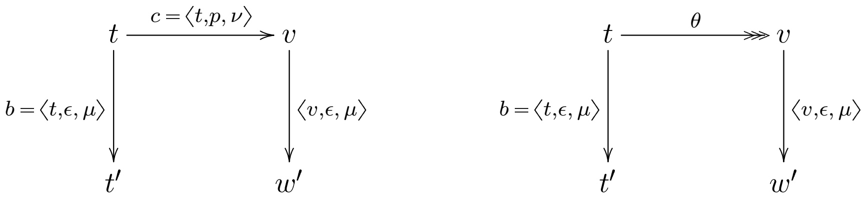
We will resort to a result presented and proved in e.g. [KKSdV95] and [BKdV03], where it is called *Strip Lemma*.[5] This result implies that whenever $t \stackrel{\gamma}{\twoheadrightarrow} t'$ and $t \xrightarrow{b} s_0$, then $t' \stackrel{b_r}{\twoheadrightarrow} s'$ and $s_0 \stackrel{\gamma_r}{\twoheadrightarrow} s'$, where b_r is the *residual* of b *after* γ.[6] The result of the lemma can be described graphically as follows:

[5]In [KKSdV90], a preliminary version of [KKSdV95], the same property is called *Parallel Moves Lemma*

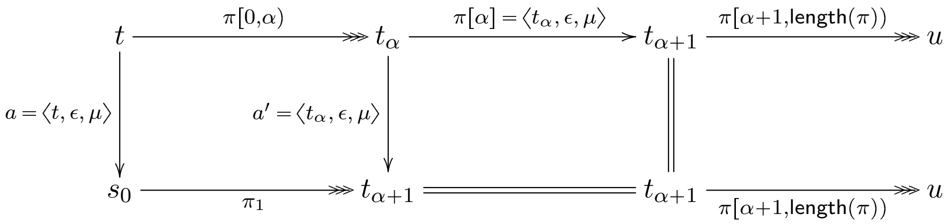
[6]The statement in [BKdV03], and also in [KKSdV90], describes also the nature of γ_r. We will not give the details here since they are not needed for this proof.



While we will not include here the formal definition of residual, we mention a feature valid for orthogonal TRSs which is crucial for this proof. Assume $b = \langle t, \epsilon, \mu \rangle$ such that $\mu : l \to h$, and $c = \langle t, p, \nu \rangle$ where $p \neq \epsilon$ and $t \xrightarrow{c} v$. Then $t = l[t_1, \ldots, t_m]$, $q \leqslant p$ for some q such that $l(q) \in \mathtt{Var}$, and therefore $v = l[t'_1, \ldots, t'_m]$. In this case, there is exactly one residual of b after c, namely $\langle v, \epsilon, \mu \rangle$. This property carries on for the residual of b after a reduction θ where $mind(\theta) > 0$, even if $\mathsf{length}(\theta)$ is a limit ordinal. Graphically:



We return to the proof. Observe that $t = l[v_1, \ldots, v_m]$ since $\langle t, \epsilon, \mu \rangle$ is a redex. Then a simple transfinite induction yields that π not including any root step would imply $u = l[v'_1, \ldots, v'_m]$, contradicting that u is a normal form. Let α be the minimum index corresponding to a root step in π. Then the described property of residuals implies that a has exactly one residual after $\pi[0, \alpha)$, which is $a' := \langle t_\alpha, \epsilon, \mu \rangle$ where t_α is the target term of $\pi[0, \alpha)$. Moreover, $\pi[\alpha]$ being a root step implies that the rule used in that step is also μ, i.e. $\pi[\alpha] = \langle t_\alpha, \epsilon, \mu \rangle = a'$. Therefore we can build the following graphic:



Hence IH on $s_0 \xrightarrow{\delta'} s$ suffices to conclude. □

Proposition 5.1.51. *Let T be a disjoint TRS which does not include collapsing rules. Then T has the property SN^∞.*[7]

Proof. First we prove the following auxiliary result: for any reduction sequence δ, limit ordinal β such that $\beta \leqslant \mathsf{length}(\delta)$, and $n < \omega$,

$$\begin{array}{rl} \text{if} & \exists \beta_1 < \beta \;/\; \forall i \; (\beta_1 < i < \beta \text{ implies } d(\delta[i]) \geqslant n) \\ \text{then} & \exists \beta' < \beta \;/\; \forall i \; (\beta' < i' < \beta \text{ implies } d(\delta[i']) > n) \end{array} \tag{5.4}$$

Assume for any δ, β and n that the premise holds. The term $src(\delta[\beta]) = tgt(\delta[\beta_1, \beta))$ can include only a finite number of redexes at depth n. Additionally, the hypothesis yields that any reduction step included in $\delta[\beta_1, \beta)$, say $\delta[j]$, satisfies $d(\delta[j]) \geqslant n$, and moreover leaves at its redex position (cfr. Dfn. 5.1.31) a symbol not being the head

[7] We conjecture that this property can be generalised to any TRS in which the sets of **head** symbols of lhss and rhss are disjoint, with exactly the same proof. The statement restricted to disjoint TRS we give here suffices for this thesis.

symbol of a left-hand side, since T is disjoint and it does not include collapsing rules. Therefore, no redex occurrence can be created at depth n, implying that any reduction step at depth exactly n included in $\delta[\beta_1, \beta)$ must correspond to a redex occurrence already included in $src(\delta[\beta_1])$ and being at the same position. Consequently, if we call k the number of steps at depth exactly n included in $\delta[\beta_1, \beta)$, we obtain $k < \omega$. Thus we conclude the proof of the auxiliary result by taking β' to be the ordinal such that $\delta[\beta']$ is the last of such steps if $k > 0$, and $\beta' := \beta_1$ if $k = 0$.

Now we prove, for any reduction sequence δ in T, that δ is convergent; i.e. that for any $n < \omega$ and β limit ordinal such that $\beta \leqslant \mathsf{length}(\delta)$,

$$\exists \beta' < \beta \ / \ \forall i \ (\beta' < i < \beta \text{ implies } d(\delta[i]) > n) \tag{5.5}$$

We conclude the proof of the proposition by proving (5.5) by induction on n. If $n = 0$, then the premise of (5.4) holds taking $\beta_1 = 0$, then we conclude by (5.4). If $n > 0$, then the premise of (5.4) holds for some β_1 by IH of (5.5) considering $n-1$ instead of n, then we conclude again by (5.4). □

5.2 Infinitary proof terms

This section is devoted to define the set of infinitary proof terms for a left-linear iTRS T, and to give some of the basic properties of proof terms. Proof terms for finitary, left-linear TRS are introduced in Section 2.2.2.

The signature for infinitary proof terms is the same as for the finitary ones, cfr. Dfn. 2.2.4; it is the result of adding the rule symbols and the concatenation symbol, i.e. the *dot*, to the signature of the object TRS.[8] Also analogously to the finitary case, not all the infinitary terms in the extended signature are valid proof terms, and the restrictions derive from conditions imposed to the occurrences of the dot. Besides the *coherence* condition described for finitary proof terms (cfr. Dfn. 2.2.5), which also applies to infinitary ones, an additional condition is needed: left components of concatenations must denote *convergent* reductions. This added condition reflects the convergence condition in the definition of infinitary reduction sequences, cfr. Dfn. 5.1.32. Similarly as the coherence condition implies that proof terms must be defined simultaneously with their source and target terms, the added convergence condition forces other notions to be defined simultaneously as well, resulting in the extensive definition we give in the following.

The definition of the set of proof terms is given in two separate stages. First, the proof terms without occurrences of the dot are introduced, along with all the needed auxiliary notions, in Section 5.2.1. We call these proof terms *infinitary multisteps*, because they denote the simultaneous contraction of coinitial sets of steps, called *multisteps* in [BKdV03], Dfn. 4.5.11.[9] Notice that *infinite* sets of coinitial steps must be considered. Subsequently, we define the whole set of valid proof terms in Section 5.2.2, by specifying the conditions which apply to the occurrences of the dot. The concatenation of an infinite number of reductions is dealt with by an ad-hoc formation rule; this allows to give a definition of the set of infinitary proof terms based on *transfinite induction*. We verify the soundness of the given definition in Section 5.2.3, and provide an alternative

[8]Recall Notation 2.2.7 for the meaning of "object TRS".

[9]We also describe the notion of multistep using the ARS model, cfr. Section 3.1.1.

principle to reason by induction on the set of proof terms in Section 5.2.4. Finally, we give some basic properties of proof terms in Section 5.2.5.

5.2.1 Infinitary multisteps

In this section, we define the set of *infinitary multisteps*, along with some basic features of a multistep, namely: how to determine its *source* and *target* terms, whether it is *convergent* or not, and its *minimum activity depth.* These concepts are needed to properly define the restrictions to be imposed to occurrences of the dot in the general definition of the set of proof terms. We give all the indicated definitions, and afterwards, some examples illustrating them.

Definition 5.2.1 (Signature for multisteps)**.** *Let $T = \langle \Sigma, R \rangle$ be a (either finitary or infinitary) TRS. We define the signature for the infinitary multisteps over T as follows:* $\Sigma^R := \Sigma \cup \{\mu/n \; / \; \mu : l \to r \in R \wedge |FV(l)| = n\}$.

Analogously to the case of finitary proof terms (cfr. Dfn. 2.2.5), all terms not including occurrences of the dot are valid proof terms.

Definition 5.2.2 (Infinitary multisteps)**.** *The set of infinitary multisteps for an iTRS $T\langle \Sigma, R \rangle$ is exactly the set of the closed (cfr. Dfn.5.1.11) terms*[10] *in $Ter^\infty(\Sigma^R)$.*

To define the source and target terms of a multistep, we define 'companion' ad-hoc iTRSs; cfr. the beginning of Section 5.1.6.

Definition 5.2.3 (src_T, tgt_T)**.** *Let $T = \langle \Sigma, R \rangle$ be a (either finitary or infinitary) TRS. We define the TRSs src_T and tgt_T as follows. The signature of both src_T and tgt_T is Σ^R. The rules of src_T are $\{\mu(x_1, \ldots, x_n) \to l[x_1, \ldots, x_n] \; / \; \mu : l \to r \in R\}$. The rules of tgt_T are $\{\mu(x_1, \ldots, x_n) \to r[x_1, \ldots, x_n] \; / \; \mu : l \to r \in R\}$.*

We remark that for any object TRS T, both src_T and tgt_T are orthogonal and disjoint; moreover, src_T does not include collapsing rules, since the lhs of a rewrite rule cannot be a variable (cfr. Dfn. 5.1.30). Therefore, both src_T and tgt_T enjoy the property UN^∞ (cfr. the comment about UN^∞ at the beginning of Section 5.1.6) and src_T enjoys also SN^∞ (cfr. Prop. 5.1.51). Moreover, given an infinitary multistep ψ, each rule symbol occurrence in ψ implies the existence of a reduction step w.r.t. *each of* src_T and tgt_T having ψ as source, so that ψ can be the source of one, or several, reduction sequences for each of these TRSs. Consequently, any infinitary multistep has exactly one src_T-normal form, and at most one tgt_T-normal form. These observations entail the soundness of the following definition.

Definition 5.2.4 (Source and target of an infinitary multistep)**.** *Let ψ be an infinitary multistep. We define $src(\psi)$ to be the src_T-normal form of ψ. Moreover, if ψ is weakly normalising w.r.t. tgt_T, then we define $tgt(\psi)$ to be the corresponding normal form; otherwise, $tgt(\psi)$ is undefined.*

For the kind of contraction activity we intend to denote with infinitary multisteps, it is correct to identify convergence with existence of target. Formally:

[10] By restricting infinitary multisteps, and later proof terms (cfr. Sec. 5.2) to be closed terms, we follow the idea expressed in [BKdV03], Remark 8.2.21 (pg. 324): "Since here we are interested in permutation equivalence, we may simply assume that reductions/proof terms are closed.". This decision simplifies, indeed, our treatment of permutation equivalence given in Sec. 5.3.

Definition 5.2.5 (Convergent infinitary multisteps)**.** *An infinitary multistep ψ is convergent iff $tgt_T(\psi)$ is defined.*

Definition 5.2.6 (Minimum activity depth of an infinitary multistep)**.** *Let ψ be an infinitary multistep. We define the minimum activity depth of ψ, notation $mind(\psi)$, as follows.*
If ψ does not include occurrences of rule symbols, i.e. if it is a term in $Ter^{\infty}(\Sigma)$, then $mind(\psi) := \omega$.
Otherwise $mind(\psi)$ is the minimum n such that exists at least one position p verifying $\psi(p) = \mu$ where μ is a rule symbol, and $n = |p|$. This case admits an equivalent inductive definition based on Notation 5.1.8:

$$\begin{array}{lcl} mind(f(\psi_1 \dots \psi_n)) & := & 1 + min(mind(\psi_1) \dots mind(\psi_n)) \\ mind(\mu(\psi_1 \dots \psi_n)) & := & 0 \end{array}$$

In the following, we will give some examples of infinitary multisteps. We will consider the following object rules: $\rho : h(g(x), y) \to k(y)$, $\tau : i(x) \to x$, $\pi : a \to b$, $\mu : f(x) \to g(x)$, $\kappa : m(x) \to h(x, x)$. Then the rules of the companion iTRSs are
src_T: $\rho(x, y) \to h(g(x), y) \quad \tau(x) \to i(x) \quad \pi \to a \quad \mu(x) \to f(x) \quad \kappa(x) \to m(x)$
tgt_T: $\rho(x, y) \to k(y) \quad \tau(x) \to x \quad \pi \to b \quad \mu(x) \to g(x) \quad \kappa(x) \to h(x, x)$

For each example, we show the source term, underlining the head symbols of some of its redexes, and the infinitary multistep denoting contraction of underlined redexes. Then we develop the computation of the source and target terms, according to Dfn. 5.2.4. To keep notation compact, we omit some parenthesis for unary symbols.

- The infinitary multistep corresponding to $j(\underline{h}(g\underline{a}, n\underline{f}b))$ is $\psi_1 := j(\rho(\pi, n\mu b))$. Computations of $src(\psi_1)$ and $tgt(\psi_1)$ follow:
  $\psi_1 = j(\rho(\pi, n\mu b)) \underset{src_T}{\longrightarrow} j(h(g\pi, n\mu b)) \underset{src_T}{\longrightarrow} j(h(ga, n\mu b)) \underset{src_T}{\longrightarrow} j(h(ga, nfb))$
  $\psi_1 = j(\rho(\pi, n\mu b)) \underset{tgt_T}{\longrightarrow} jkn\mu b \underset{tgt_T}{\longrightarrow} jkngb.$
- $\psi_2 := \kappa(\mu(a))$ corresponds to $\underline{m}(\underline{f}(a))$. We compute the source and target terms:
  $\psi_2 = \kappa(\mu(a)) \underset{src_T}{\longrightarrow} m(\mu(a)) \underset{src_T}{\longrightarrow} m(f(a))$
  $\psi_2 = \kappa(\mu(a)) \underset{tgt_T}{\longrightarrow} h(\mu(a), \mu(a)) \underset{tgt_T}{\longrightarrow} h(g(a), \mu(a)) \underset{tgt_T}{\longrightarrow} h(g(a), g(a)).$
- $\psi_3 := \mu^{\omega}$ corresponds to $\underline{f}^{\omega}$. Let us compute source and target:
  $\psi_3 = \mu^{\omega} \underset{src_T}{\longrightarrow} f(\mu^{\omega}) \underset{src_T}{\longrightarrow} f(f(\mu^{\omega})) \underset{src_T}{\twoheadrightarrow} f^{\omega}$
  $\psi_3 = \mu^{\omega} \underset{tgt_T}{\longrightarrow} g(\mu^{\omega}) \underset{tgt_T}{\longrightarrow} g(g(\mu^{\omega})) \underset{tgt_T}{\twoheadrightarrow} g^{\omega}.$
- $\psi_4 := \tau^{\omega}$ corresponds to $\underline{i}^{\omega}$.
  The computation of source runs as in the previous case: $\psi_4 = \tau^{\omega} \underset{src_T}{\twoheadrightarrow} i^{\omega}$. On the other hand, the target of all tgt_T redex occurrences in τ^{ω} (namely, $\langle 1^i, \tau(x) \to x, \{x \to \tau^{\omega}\}\rangle$) is again τ^{ω}. Therefore $tgt(\psi_4)$ is undefined.
- Finally, $\psi_5 = j(\rho(\tau^{\omega}, \pi))$ corresponds to $j(\underline{h}(g\underline{i}^{\omega}, \underline{a}))$.
  Computation of source follows:
  $\psi_5 = j(\rho(\tau^{\omega}, \pi)) \underset{src_T}{\longrightarrow} j(h(g\tau^{\omega}, \pi)) \underset{src_T}{\longrightarrow} j(h(g\tau^{\omega}, a)) \underset{src_T}{\twoheadrightarrow} j(h(gi^{\omega}, a))).$
  Many tgt_T reduction sequences from ψ_5 are possible, e.g.:
  $\psi_5 = j(\rho(\tau^{\omega}, \pi)) \underset{tgt_T}{\longrightarrow} jk\pi \underset{tgt_T}{\longrightarrow} jkb$

$\psi_5 = j(\rho(\tau^\omega, \pi)) \underset{tgt_T}{\longrightarrow} j(\rho(\tau^\omega, b)) \underset{tgt_T}{\twoheadrightarrow} j(\rho(\tau^\omega, b)) \underset{tgt_T}{\longrightarrow} jkb$ where the i-th step for $1 \leqslant i < \omega$ is $\langle j(\rho(\tau^\omega, b)), 11 \cdot 1^i, \tau(x) \to x, \{x := \tau^\omega\}\rangle$
$\psi_5 = j(\rho(\tau^\omega, b)) \underset{tgt_T}{\twoheadrightarrow} j(\rho(\tau^\omega, b))$ where all steps are $\langle \psi_5, 11, \tau(x) \to x, \{x := \tau^\omega\}\rangle$, a divergent tgt_T reduction sequence.
Then ψ_5 admit both convergent and divergent reduction sequences in tgt_T. As ψ_5 is tgt_T-weakly normalising, we get $tgt(\psi_5) = jkb$.

5.2.2 The whole set of proof terms

In this section we give the definition of the set of all valid infinitary proof terms, by providing precise rules for the inclusion of the occurrences of the concatenation symbol, that is, the *dot*. The foundation for this definition is given by the set of infinitary multisteps, defined in the previous section.

As pointed out in the introduction to Section 5.2, two conditions apply for $\psi \cdot \phi$ to be a valid proof term. First, the activity denoted by ψ must be *convergent*, i.e., it should exist at least one way to render such activity as a convergent reduction sequence; this condition implies particularly that the target term of ψ can be uniquely determined. Second, the activity denoted by ψ must be *coherent* with that of ϕ in the following sense: the target term of (the activity denoted by) ψ must coincide with the source term of (that corresponding to) ϕ.

The need of imposing such conditions on the occurrences of the dot implies that the set of proof terms must be defined along with the source, target and convergence condition for each proof term, in a joint definition. Convergence depends in turn of the depth of the contraction activity being denoted by a proof term; therefore, *minimum activity depth* of proof terms must be merged within the same, extensive definition.

The set of infinitary proof terms is defined by an *inductive* construction, where the base case is given by the infinitary multisteps, and inductive rules govern the addition of dots. A *binary concatenation* rule allows proof terms of the form $\psi \cdot \phi$, given that ψ and ϕ are proof terms. Note that some mechanism must be provided to denote the concatenation of an *infinite* series of reduction sequences, or more generally of contraction activities. The definition to be presented in the following admits terms including an infinite number of occurrences of the dot. These *infinite concatenations* are defined by a separate rule, different than that allowing to define binary concatenations. In the infinite concatenation rule, special care is taken to guarantee that no component is "lost", i.e., that the root of any component is at a finite distance from the root in the corresponding proof term.

The separate rules for binary and infinite concatenation give rise to potential ambiguities in the construction of a proof term. To avoid the possibility of such ambiguities, the definition of the set of proof terms is *layered*, such that the proof terms included in a layer can be built taking as components proof terms in previous layers only. Countable ordinals are used as layers for proof terms, and each proof term belongs to exactly one layer. The separation of proof terms in layers yields also a *transfinite induction principle* to reason about the set of infinitary proof terms. The base layer corresponds to infinitary multisteps, and the layers for limit ordinals correspond exactly to infinite concatenations. A second, alternative induction principle for the set of proof terms is introduced in Section 5.2.4.

The aforementioned considerations lead to the following definitions.

Definition 5.2.7 (Signature for proof terms)**.** *Let $T = \langle \Sigma, R \rangle$ be a (either finitary or infinitary) TRS. We define the signature for the proof terms over T as follows: $\Sigma^{PT} := \Sigma^R \cup \{\cdot/2\}$. Recall the definition of Σ^R, cfr. Dfn. 5.2.1.*

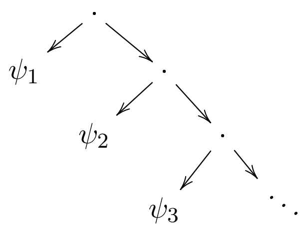
Note that the signature for infinitary proof terms coincide with that of finitary ones, cfr. Dfn. 2.2.4.

Definition 5.2.8 ($\mathbf{PT}_\alpha$, set of proof terms at layer α)**.** *Let T be a TRS, and α a countable ordinal. We define $\mathbf{PT}_\alpha$, the α-th layer in the construction of the set of proof terms for T, along with the source, target, convergence condition, and minimum activity depth of any proof term in $\mathbf{PT}_\alpha$. If $\psi \in \mathbf{PT}_\alpha$, we will write $src(\psi)$, $tgt(\psi)$ and $mind(\psi)$ for the source, target and minimum activity depth of ψ respectively.*

If $\alpha = 0$, then $\mathbf{PT}_\alpha := \varnothing$. Otherwise, we proceed inductively on α, defining $\mathbf{PT}_\alpha$ to be the smallest set in $Ter^\infty(\Sigma^{PT})$ verifying the following conditions.

1. *If $\alpha = 1$ and ψ is an infinitary multistep for T, then $\psi \in \mathbf{PT}_\alpha$. The source, target, convergence condition and minimum activity depth of ψ coincide with the definitions given for infinitary multisteps in Sec. 5.2.1.*

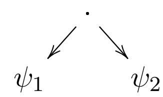
2. *Assume that for any $i < \omega$, $\psi_i \in \mathbf{PT}_{\alpha_i}$, such that $\alpha = \sum_{i<\omega} \alpha_i$; cfr. Dfn. 5.1.1. Moreover, assume that for all n, ψ_n is convergent, and $tgt(\psi_n) = src(\psi_{n+1})$.*
   Then $\psi := \langle P, F \rangle \in \mathbf{PT}_\alpha$, where $P := \{2^n \ / \ n < w\} \cup (\bigcup_{n<\omega} 2^n 1 \cdot \mathtt{Pos}(\psi_n))$, $F(2^n) := \cdot$, and $F(2^n 1p) := \psi_n(p)$. A concise term notation for ψ is $\cdot_{i<\omega}\ \psi_i$; being in fact an abbreviation for $\psi_1 \cdot (\psi_2 \cdot (\psi_3 \cdot \ldots))$.
   A graphical representation is



   We define the source, target and minimum activity depth of ψ as follows: $src(\psi) := src(\psi_0)$, $tgt(\psi) := \lim_{i\to\omega} tgt(\psi_i)$ and $mind(\psi) := min(mind(\psi_i)_{i<\omega})$; notice that $tgt(\psi)$ can be undefined. We define that ψ is convergent iff for all $k < \omega$, there is some $n < \omega$ such that $mind(\psi_j) > k$ if $j > n$.

3. *Assume that $\psi_1 \in \mathbf{PT}_{\alpha_1}$, $\psi_2 \in \mathbf{PT}_{\alpha_2}$, α_2 is a successor ordinal, ψ_1 is convergent, $tgt(\psi_1) = src(\psi_2)$, and $\alpha = \alpha_1 + \alpha_2 + 1$. Then $\psi := \langle P, F \rangle \in \mathbf{PT}_\alpha$, where $P := \{\epsilon\} \cup (1 \cdot \mathtt{Pos}(\psi_1)) \cup (2 \cdot \mathtt{Pos}(\psi_2))$, $F(\epsilon) := \cdot$, and $F(ip) := \psi_i(p)$ for $i = 1, 2$.*

   A concise term notation for ψ is $\psi_1 \cdot \psi_2$. A graphical notation is



   If $\psi = \psi_1 \cdot \psi_2$, then we define $src(\psi) := src(\psi_1)$, $tgt(\psi) := tgt(\psi_2)$ and $mind(\psi) = min(mind(\psi_1), mind(\psi_2))$; ψ is convergent iff ψ_2 is.

4. *Assume that $\psi_i \in \mathbf{PT}_{\alpha_i}$ for $i = 1, 2, \ldots, n$, that $\alpha_i > 1$ for at least one i, $f/n \in \Sigma$ (resp. μ/n is a rule symbol), and $\alpha = \alpha_1 + \ldots + \alpha_n + 1$. Then $\psi := \langle P, F \rangle \in \mathbf{PT}_\alpha$, where $P := \{\epsilon\} \cup (\bigcup_{1 \leqslant i \leqslant n} i \cdot \mathtt{Pos}(\psi_i))$, $F(\epsilon) := f$ (resp. $F(\epsilon) := \mu$), and $F(ip) := \psi_i(p)$ for $i = 1, 2, \ldots, n$.*

   A concise term notation for ψ is $f(\psi_1, \ldots, \psi_n)$ (resp. $\mu(\psi_1, \ldots, \psi_n)$).

If $f \in \Sigma$, i.e. it is an object symbol, we define $src(\psi) = f(src(\psi_1), \ldots, src(\psi_n))$, $tgt(\psi) = f(tgt(\psi_1), \ldots, tgt(\psi_n))$, $mind(\psi) := 1 + min(mind(\psi_1), \ldots, mind(\psi_n))$. In this case, ψ is convergent iff all ψ_i are. We observe that $tgt(\psi)$ is undefined if at least one $tgt(\psi_i)$ is.

If μ is a rule symbol such that $\mu : l \to r$, we define $src(\psi) = l[src(\psi_1), \ldots, src(\psi_n)]$, $tgt(\psi) = r[tgt(\psi_1), \ldots, tgt(\psi_n)]$, and $mind(\psi) := 0$. In this case, ψ is convergent iff all ψ_i ***corresponding to some*** *x_i* ***occurring in*** *r are. We observe that $tgt(\psi)$ is undefined if at least one $tgt(\psi_i)$ is, for the ψ_i already mentioned.*

Definition 5.2.9 (**PT**, the set of proof terms)**.** *We define the set of proof terms as follows:* $\mathbf{PT} := \bigcup_{\alpha<\omega_1} \mathbf{PT}_\alpha$.

We notice that all proof terms are *closed* terms in $Ter^\infty(\Sigma^{PT})$. This fact is a consequence of the definition of the set of infinitary multisteps, which are the base layer in the definition of **PT**. Cfr. the footnote on Dfn. 5.2.2.

We will say that a proof term ψ is an *infinite concatenation* iff $\psi(2^n) = \cdot$ for all $n < \omega$. Observe that all infinite concatenations admit the concise term notation $\psi = \cdot_{i<\omega}\, \psi_i$, where $\psi_n = \psi|_{2^n 1}$. Furthermore, ψ not being an infinite concatenation implies the existence of some $n < \omega$ such that $2^n \in \mathtt{Pos}(\psi)$ and $\psi(2^n) \neq \cdot$.

We define as *trivial* proof terms those which denote no activity.

Definition 5.2.10. *Let ψ be a proof term. We will say that ψ is a* trivial proof term *iff it does not include any rule symbol occurrences.*

We remark that the structure of trivial proof terms can be arbitrarily complex, i.e. $\cdot_{j<\omega}\, (\cdot_{i<\omega}\, a)$ is a trivial proof term. The following property of trivial proof terms is used later on in this chapter.

Lemma 5.2.11. *Let ψ be a proof term. Then ψ is trivial iff $mind(\psi) = \omega$.*

Proof. For the $\Rightarrow$) direction, a straightforward induction on ψ (i.e. on α such that $\psi \in \mathbf{PT}_\alpha$) suffices. For the base case, i.e. when ψ is an infinitary multistep, we just refer to Dfn. 5.2.6.

For the $\Leftarrow$) direction, a similar induction on ψ yields the contrapositive, i.e. that if ψ includes at least one rule symbol occurrence, then $mind(\psi) < \omega$. If ψ is an infinitary multistep, then we define n to be the least depth of a rule symbol occurrence in ψ. An easy induction on n yields $mind(\psi) = n$. If $\psi = \mu(\psi_1, \ldots, \psi_m)$, then $mind(\psi) = 0$. For the other cases, IH suffices to conclude. □

We observe that all the *finitary proof terms* are comprised in Dfn. 5.2.9, and moreover that they are convergent and correspond to finite layers. This can be verified by means of a simple inductive argument over Dfn. 2.2.5. Therefore, all the examples given in Section 2.2.2 correspond to infinitary proof terms, with the same source and target terms. The set of *infinitary multisteps* is included in **PT**, hence the examples given at the end of Section 5.2.1 are infinitary proof terms as well. We give some additional examples of infinitary proof terms, using the rules $\mu : f(x) \to g(x)$ and $\nu : g(x) \to k(x)$. We refer to the formation rules in Dfn. 5.2.8.

Consider the reduction sequence $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega$ having length ω. The i-th step of this sequence, namely $g^i(f^\omega) \to g^{i+1}(f^\omega)$, can be described by the proof term $g^i(\mu(f^\omega))$. It is straightforward to check that the sequence formed by these proof terms verifies the conditions of the infinitary composition rule, and that the depth of the denoted activity tends to infinity. Therefore $\cdot_{i<\omega}\, g^i(\mu(f^\omega))$ is a valid proof term, by means of rule 2; the indicated condition about depths implies that it is moreover a *convergent* proof term. We observe that $src(\cdot_{i<\omega}\, g^i(\mu(f^\omega))) = src(\mu(f^\omega)) = f^\omega$. In order to obtain $tgt(\cdot_{i<\omega}\, g^i(\mu(f^\omega))) = g^\omega$, it is enough to observe that the sequence of targets of each $g^i(\mu(f^\omega))$, namely $g(f^\omega), g^2(f^\omega), \ldots$, converges to that term.

Analogously, the reduction sequence $f^\omega \to g(f^\omega) \to k(f^\omega) \to k(g(f^\omega)) \to k^2(f^\omega) \twoheadrightarrow k^\omega$ can be denoted by either $\cdot_{i<\omega}\, (k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega)))$ or $\cdot_{i<\omega}\, (k^i(\mu(f^\omega) \cdot \nu(f^\omega)))$, again by means of rule 2. In the latter case, for any $n < \omega$, we obtain that $\mu(f^\omega) \cdot \nu(f^\omega)$ is a valid proof term by rule 3; therefore, applying n times rule 4 we get $k^n(\mu(f^\omega) \cdot \nu(f^\omega))$.

In turn, the reduction sequence $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega \to k(g^\omega)$ can be denoted by the proof term $(\cdot_{i<\omega}\, g^i(\mu(f^\omega))) \cdot \nu(g^\omega)$ by means of rule 3, because $\cdot_{i<\omega}\, g^i(\mu(f^\omega))$ is a convergent proof term, and $tgt(\cdot_{i<\omega}\, g^i(\mu(f^\omega))) = src(\nu(g^\omega)) = g^\omega$. We obtain $src((\cdot_{i<\omega}\, g^i(\mu(f^\omega))) \cdot \nu(g^\omega)) = src(\cdot_{i<\omega}\, g^i(\mu(f^\omega))) = f^\omega$, and $tgt((\cdot_{i<\omega}\, g^i(\mu(f^\omega))) \cdot \nu(g^\omega)) = tgt(\nu(g^\omega)) = k(g^\omega)$.

As observed for finitary proof terms in Section 2.2.2, the rules defining the set of proof terms can be combined in different ways. A simple example follows: rule 4 implies that $k(\cdot_{i<\omega}\, g^i(\mu(f^\omega)))$ is a valid proof term, given that $\cdot_{i<\omega}\, g^i(\mu(f^\omega))$ is, as we have already verified. We get $src(k(\cdot_{i<\omega}\, g^i(\mu(f^\omega)))) = k(src(\cdot_{i<\omega}\, g^i(\mu(f^\omega)))) = k(f^\omega)$, and analogously for the target term. The reduction $f^\omega \twoheadrightarrow g^\omega \twoheadrightarrow k^\omega$, can be denoted by either $\cdot_{i<\omega}\, g^i(\mu(f^\omega)) \cdot \cdot_{i<\omega}\, k^i(\nu(g^\omega))$ (if taken as a sequence having length $\omega \times 2$) or $\mu^\omega \cdot \nu^\omega$ (if considered as the composition of two infinite simultaneous reductions), in both cases by means of rule 3, since both $\cdot_{i<\omega}\, g^i(\mu(f^\omega))$ and μ^ω are convergent. Specifically, $\cdot_{i<\omega}\, g^i(\mu(f^\omega)) \cdot \cdot_{i<\omega}\, k^i(\nu(g^\omega)) \in \mathbf{PT}_{\omega*2+1}$, and $\mu^\omega \cdot \nu^\omega \in \mathbf{PT}_3$. The reduction $f^\omega \to f(g(f^\omega)) \to f(k(f^\omega)) \to f(k(g(f^\omega))) \to f(k^2(f^\omega)) \twoheadrightarrow f(k^\omega) \to g(k^\omega)$ can be denoted by $\cdot_{i<\omega}\, f(k^i(\mu(f^\omega) \cdot \nu(f^\omega))) \cdot \mu(k^\omega)$, and also by $f(\cdot_{i<\omega}\, k^i(\mu(f^\omega) \cdot \nu(f^\omega))) \cdot \mu(k^\omega)$.

We give some examples involving non-convergent proof terms; we use the rules $\tau : i(x) \to i(x)$ and $\rho : h(x,y) \to m(x)$. The proof term $\tau(a)$ is convergent; $i(a)$ is both its source and target term. Therefore, rule 2 implies that $\cdot_{i<\omega}\, \tau(a)$ is a valid proof term. In turn, the minimum activity depth of all the components of $\cdot_{i<\omega}\, \tau(a)$ is $mind(\tau(a)) = 0$, so that $\cdot_{i<\omega}\, \tau(a)$ is not convergent. Notice that $src(\cdot_{i<\omega}\, \tau(a))$ is well defined, namely, it is $src(\tau(a)) = i(a)$.

Non-convergence of $\cdot_{i<\omega}\, \tau(a)$ implies that neither $\nu(\cdot_{i<\omega}\, \tau(a))$ nor $\rho(\cdot_{i<\omega}\, \tau(a), \mu(b))$ are convergent. On the other hand, $\rho(\mu(b), \cdot_{i<\omega}\, \tau(a))$ is a convergent proof term; observe that the variable replaced by the non-convergent subterm $\cdot_{i<\omega}\, \tau(a)$ in the left-hand side of ρ, namely y, does not occur in the corresponding right-hand side. Computation of the source and target terms yields $src(\rho(\mu(b), \cdot_{i<\omega}\, \tau(a))) = h(src(\mu(b), src(\cdot_{i<\omega}\, \tau(a))) = h(f(b), i(a))$, and $tgt(\rho(\mu(b), \cdot_{i<\omega}\, \tau(a))) = m(tgt(\mu(b))) = m(g(b))$.

Finally, we remark that infinite composition can be combined with itself. Let us consider a reduction sequence having length ω^2, and ϕ_{ij} be a proof term denoting its $\omega * i + j$-th step, so that for each $i < \omega$, $\cdot_{j<\omega}\, \phi_{ij}$ denotes the subsequence including the steps from the $\omega * i$-th up to the $\omega * (i+1)$-th. Then $\cdot_{i<\omega}\, \cdot_{j<\omega}\, \phi_{ij}$ is a proof term denoting the entire reduction sequence. By iteration of this pattern, proof terms can be built denoting reduction sequences of any countable ordinal length. This claim is proved in Sec. 5.4.

5.2.3 Soundness of the definitions

In this section, we study the definition of the set of valid proof terms in some detail, stating and proving properties related to its soundness.

Lemma 5.2.12. *Let ψ, α such that $\psi \in \mathbf{PT}_\alpha$. Then ψ is an infinite concatenation iff α is a limit ordinal iff ψ is generated by case 2 in Dfn. 5.2.8.*

Proof. We proceed by induction on α, analysing the rules in Dfn. 5.2.8.
Case 1: in this case ψ is an infinitary multistep, so that $\psi(2^0) = \psi(\epsilon) \neq \cdot$.
Case 2: in this case $\psi = \cdot_{i<\omega} \psi_i$, that is, an infinite concatenation. It is enough to observe that $\mathbf{PT}_0 = \varnothing$, and that $\alpha_i > 0$ for all i implies that $\sum_{i<\omega} \alpha_i$ is a limit ordinal.
Case 3: in this case $\psi = \psi_1 \cdot \psi_2$ where $\psi_i \in \mathbf{PT}_{\alpha_i}$, α_2 is a successor ordinal, and $\alpha = \alpha_1 + \alpha_2 + 1$, i.e. a successor ordinal. IH on ψ_2 implies that $\psi_2(2^n) \neq \cdot$ for some $n < \omega$. We conclude by observing that $\psi(2^{n+1}) = \psi_2(2^n)$.
Case 4: in this case it is immediate that $\psi(2^0) = \psi(\epsilon) \neq \cdot$, and that α is a successor ordinal. □

Lemma 5.2.13. *Let ψ, α such that $\psi \in \mathbf{PT}_\alpha$. Then ψ is an infinitary multistep iff $\alpha = 1$ iff ψ is generated by case 1 in Dfn. 5.2.8.*

Proof. We proceed by induction on α, analysing the rules in Dfn. 5.2.8.
Case 1: we conclude immediately.
Case 2: in this case ψ is not an infinitary multistep, observe e.g. that $\psi(\epsilon) = \cdot$, and α is a limit ordinal, cfr. Lem. 5.2.12. Thus we conclude.
Case 3: in this case ψ is not an infinitary multistep, observe e.g. that $\psi(\epsilon) = \cdot$, and $\alpha > \alpha_1 + 1 > 1$, recall $\mathbf{PT}_0 = \varnothing$. Thus we conclude.
Case 4: in this case $\psi = f(\psi_1, \ldots, \psi_n)$ where $\psi_i \in \mathbf{PT}_{\alpha_i}$ for all i, and exists some k such that $\alpha_k > 1$. Observe that $\alpha > \alpha_k > 1$, then we can apply IH to obtain that ψ_k is not an infinitary multistep, hence ψ is neither. Thus we conclude. □

The set $\mathbf{PT}$ is closed by operations, formally:

Proposition 5.2.14 (Completeness of $\mathbf{PT}$)**.**

1. *If ψ is an infinite multistep, then $\psi \in \mathbf{PT}$.*
2. *If $\psi_1, \psi_2 \in \mathbf{PT}$, ψ_1 is convergent, and $src(\psi_2) = tgt(\psi_1)$, then $\psi_1 \cdot \psi_2 \in \mathbf{PT}$.*
3. *Given a sequence $\langle \psi_i \rangle_{i<\omega}$ such that for all i, $\psi_i \in \mathbf{PT}$, ψ_i are convergent, and $tgt(\psi_i) = src(\psi_{i+1})$, then $\cdot_{i<\omega} \psi_i \in \mathbf{PT}$.*
4. *If $\psi_1, \ldots, \psi_n \in \mathbf{PT}$ and $f \in \Sigma$, then $f(\psi_1, \ldots, \psi_n) \in \mathbf{PT}$.*
5. *If $\psi_1, \ldots, \psi_n \in \mathbf{PT}$ and μ is a rule symbol, then $\mu(\psi_1, \ldots, \psi_n) \in \mathbf{PT}$.*

Proof. We prove each item separately, referring to cases in Dfn. 5.2.8.
Item 1: in this case $\psi \in \mathbf{PT}_1$, this is immediate from case 1.
Item 2: Let α_1, α_2 such that $\psi_i \in \mathbf{PT}_{\alpha_i}$ for $i = 1, 2$. If α_2 is a successor ordinal, then $\psi_1 \cdot \psi_2 \in \mathbf{PT}_{\alpha_1+\alpha_2+1} \subseteq \mathbf{PT}$. If α_2 is a limit ordinal, then Lem. 5.2.12 implies that $\psi_2 = \cdot_{i<\omega} \phi_i$, where for all i, ϕ_i is convergent and $tgt(\phi_i) = src(\phi_{i+1})$; cfr. case 2. On the other hand, hypotheses imply that ψ_1 is convergent and $tgt(\psi_1) = src(\psi_2) = src(\phi_0)$. Then $\psi_1 \cdot \psi_2 \in \mathbf{PT}_{\alpha_1+\alpha_2}$, again by case 2. Observe that $\psi_1 \cdot \psi_2 = \psi_1 \cdot (\cdot_{i<\omega} \phi_i) = \cdot_{i<\omega} \phi'_i$ where $\phi'_0 := \psi_1$ and $\phi'_{i+1} := \phi_i$ for all $i < \omega$.
Item 3: we conclude just by observing that case 2 implies that $\cdot_{i<\omega} \psi_i \in \mathbf{PT}_\beta$, where $\psi_i \in \mathbf{PT}_{\alpha_i}$ for all $i < \omega$ and $\beta := \sum_{i<\omega} \alpha_i$.
Item 4 and Item 5: it is enough to observe that case 4 applies. □

Now we prove uniqueness of formation, w.r.t. the layered definition, for any valid proof term.

Lemma 5.2.15. *Let $\psi \in \mathbf{PT}$. Then there exists a unique α such that $\psi \in \mathbf{PT}_\alpha$, and moreover there is exactly one case in Dfn. 5.2.8 justifying $\psi \in \mathbf{PT}_\alpha$.*

Proof. We will prove the following statement, which is equivalent to the desired result.

> Let $\psi \in \mathbf{PT}$, α minimal for $\psi \in \mathbf{PT}_\alpha$, and β such that $\psi \in \mathbf{PT}_\beta$. Then $\beta = \alpha$, and there is exactly one case in Dfn. 5.2.8 justifying $\psi \in \mathbf{PT}_\alpha$.

We proceed by induction on α, analysing which case in Dfn. 5.2.8 could justify $\psi \in \mathbf{PT}_\alpha$.
Case 1. In this case $\alpha = 1$ and ψ is an infinitary multistep. We conclude by Lem. 5.2.13.
Case 2. In this case $\psi = \cdot_{i<\omega} \psi_i$ such that $\psi_i \in \mathbf{PT}_{\alpha_i}$ and $\alpha = \sum_{i<\omega} \alpha_i$. Observe that $\alpha > \alpha_i$ for all i, recall $\mathbf{PT}_0 = \varnothing$. Assume $\psi \in \mathbf{PT}_\beta$. Lem. 5.2.12 implies that this assertion is generated by case 2, implying that $\beta = \sum_{i<\omega} \beta_i$ and $\psi_i \in \mathbf{PT}_{\beta_i}$. Let $i < \omega$ and γ_i minimal for $\psi_i \in \mathbf{PT}_{\gamma_i}$. Then $\gamma_i \leqslant \alpha_i < \alpha$, and therefore IH can be applied twice on each ψ_i obtaining $\beta_i = \alpha_i = \gamma_i$. Thus we conclude.
Case 3. In this case $\psi = \psi_1 \cdot \psi_2$, $\alpha = \alpha_1 + \alpha_2 + 1$, α_2 is a successor ordinal, and $\psi_i \in \mathbf{PT}_{\alpha_i}$ for $i = 1, 2$. Then Lem. 5.2.12 applied to ψ_2 implies that it is not an infinite concatenation, thus neither is ψ. On the other hand, observe that α is a successor ordinal verifying $\alpha > \alpha_i$ for $i = 1, 2$. Assume $\psi \in \mathbf{PT}_\beta$. Then applying again Lem. 5.2.12 yields that this assertion is not justified by case 2 (since ψ is not an infinite concatenation); therefore, the shape of ψ (recall $\psi(\epsilon) = \cdot$) leaves case 3 as the only valid option. Hence $\beta = \beta_1 + \beta_2 + 1$ where $\psi_i \in \mathbf{PT}_{\beta_i}$ for $i = 1, 2$. An argument analogous to that used in the previous case, i.e. resorting to the IH on each ψ_i, yields $\beta_i = \alpha_i$. Thus we conclude.
Case 4. In this case $\psi = f(\psi_1, \ldots, \psi_m)$ and $\alpha = \alpha_1 + \ldots + \alpha_m + 1$, where $\psi_i \in \mathbf{PT}_{\alpha_i}$ for all i, and exists some k verifying $\alpha_k > 1$. Then Lem. 5.2.13 implies that ψ_k is not an infinitary multistep, so that neither is ψ. Therefore, the shape of ψ (recall $\psi(\epsilon) \neq \cdot$) leaves case 4 as the only valid option, implying that $\beta = \beta_1 + \ldots + \beta_m + 1$ where $\psi_i \in \mathbf{PT}_{\beta_i}$ for all i. We conclude by obtaining $\beta_i = \alpha_i$ through an argument resorting to the IH, like in the previous cases. □

We remark that Lem. 5.2.15 ensures that transfinite induction on the layer attached to each proof term, combined with rule analysis w.r.t. Dfn. 5.2.8, is a sound principle to reason about the set of proof terms.

5.2.4 An alternative induction principle

As noted in Section 5.2.2, the principle given by the layered definition of **PT** allows to perform reasonings by transfinite induction over the set of infinitary proof terms. In this section we introduce a second sound induction principle for proof terms, based in their concise notation. This induction principle is equivalent to that given by layers. Some of the forthcoming proofs about proof terms resort to this alternative, equivalent induction principle, while other proceed by transfinite induction on the layer attached to each proof term. The intent is to obtain proofs as intuitively simple as possible, without compromising their validity. The following proposition introduces the alternative induction principle, and shows that it is equivalent to that given by layers.

Proposition 5.2.16 (Alternative, equivalent induction principle for **PT**). *Let P a unary predicate satisfying all the following conditions:*

1. *If ψ is an infinitary multistep, then $P(\psi)$ holds.*
2. *For all ψ_1, ψ_2 such that $\psi_1 \cdot \psi_2 \in \mathbf{PT}$, $P(\psi_1)$ and $P(\psi_2)$ imply $P(\psi_1 \cdot \psi_2)$.*
3. *Given $\langle \psi_i \rangle_{i<\omega}$ such that $\cdot_{i<\omega}\, \psi_i \in \mathbf{PT}$, $P(\psi_i)$ for all i imply $P(\cdot_{i<\omega}\, \psi_i)$.*
4. *For all $\psi_1, \ldots, \psi_n \in \mathbf{PT}$ and for all $f \in \Sigma$, $P(\psi_1), \ldots, P(\psi_n)$ imply $P(f(\psi_1, \ldots, \psi_n))$.*
5. *For all $\psi_1, \ldots, \psi_n \in \mathbf{PT}$ and for any rule symbol μ, $P(\psi_1), \ldots, P(\psi_n)$ imply $P(\mu(\psi_1, \ldots, \psi_n))$.*

Then $P(\psi)$ holds for all $\psi \in \mathbf{PT}$.

Proof. We proceed by induction on α where $\psi \in \mathbf{PT}_\alpha$, referring to the conditions in the lemma statement.
If $\alpha = 1$, then Lem. 5.2.13 implies ψ to be an infinitary multistep, so that we conclude by condition 1.
Assume that α is a successor ordinal. If $\psi(\epsilon) = \cdot$, then Lem 5.2.12 implies that $\psi = \psi_1 \cdot \psi_2$, such that for $i = 1, 2$, $\psi_i \in \mathbf{PT}_{\alpha_i}$ for some α_i satisfying $\alpha > \alpha_i$. Then IH can be applied on each ψ_i yielding $P(\psi_1)$ and $P(\psi_2)$ to hold. We conclude by condition 2.
Otherwise, i.e. if $\psi = f(\psi_1, \ldots, \psi_m)$ or $\psi = \mu(\psi_1, \ldots, \psi_m)$, then Lem. 5.2.13 implies that ψ is not an infinitary multistep, therefore for all i, $\psi_i \in \mathbf{PT}_{\alpha_i}$ where $\alpha > \alpha_i$. Then IH on each i yield $P(\psi_i)$ to hold for all i. We conclude by condition 4.
Assume that α is a limit ordinal. In this case, Lem 5.2.12 implies that $\psi = \cdot_{i<\omega}\, \psi_i$, such that for all $i < \omega$, $\psi_i \in \mathbf{PT}_{\alpha_i}$ where $\alpha_i < \alpha$. Then we can apply IH on each ψ_i obtaining that $P(\psi_i)$ holds for all $i < \omega$. We conclude by condition 3. □

In the proofs resorting to Prop. 5.2.16, we indicate as *induction hypotheses* the hypotheses of each case in the Proposition. E.g. when proving a property for proof terms having the form $\psi_1 \cdot \psi_2$, we will refer to the hypotheses of case 2 in Prop. 5.2.16, namely that the property holds for ψ_1 and ψ_2, as induction hypothesis in the proof.

5.2.5 Basic properties of proof terms

The following lemma shows that the target of a convergent proof term is always defined, and also a correspondence between $mind(\psi)$ and the existence of a fixed prefix for the activity denoted by ψ. These two results are merged in the same lemma because they need to be proved simultaneously.

Lemma 5.2.17. *Let ψ be a convergent proof. Then* *(a)* $tgt(\psi)$ *is defined.*

(b) For all $n < \omega$, $mind(\psi) > n$ *implies* $\mathtt{dist}(src(\psi), tgt(\psi)) < 2^{-n}$.

Proof. We proceed by induction on α where $\psi \in \mathbf{PT}_\alpha$, analysing the case in Dfn. 5.2.8 corresponding to ψ. If ψ is an infinitary multistep, then item (a) is immediate from Dfn. 5.2.5, and for item (b) an easy induction on n suffices.

Assume $\psi = \psi_1 \cdot \psi_2$. Item (a) can be proved by just applying IH on ψ_2. To obtain item (b), observe that IH applies to ψ_i for $i = 1, 2$, since $mind(\psi_i) \geqslant mind(\psi) > n$, yielding $\mathtt{dist}(src(\psi_i), tgt(\psi_i)) < 2^{-n}$. Moreover Lemma 5.1.25 implies $\mathtt{dist}(src(\psi), tgt(\psi)) \leqslant max(\mathtt{dist}(src(\psi), src(\psi_2)), \mathtt{dist}(src(\psi_2), tgt(\psi))$. Thus we conclude by observing that $src(\psi) = src(\psi_1)$, $src(\psi_2) = tgt(\psi_1)$, and $tgt(\psi) = tgt(\psi_2)$.

Assume $\psi = \cdot_{i<\omega} \psi_i$.

We prove item (a). For any $i < \omega$, ψ_i being convergent implies that IH applies to obtain that $tgt(\psi_i)$ is defined. Let $n < \omega$, and k_n such that $mind(\psi_i) > n$ if $k_n < i < \omega$. Let j such that $k_n < j$. Then IH:(b) applies on $\psi_{k_n+1} \cdot \ldots \cdot \psi_j$, implying $\mathtt{dist}(tgt(\psi_{k_n+1}), tgt(\psi_j)) < 2^{-n}$.[11] Therefore, for any position p and $j \geqslant k_{|p|} + 1$, $p \in \mathtt{Pos}(tgt(\psi_j))$ iff $p \in \mathtt{Pos}(tgt(\psi_{k_{|p|}+1}))$, and in such case, $tgt(\psi_j)(p) = tgt(\psi_{k_{|p|}+1})(p)$. We define $t = \langle P, F \rangle$ as follows: $p \in P$ iff $p \in \mathtt{Pos}(tgt(\psi_{k_{|p|}+1}))$, and $F(p) := tgt(\psi_{k_{|p|}+1})(p)$ for all $p \in P$. To conclude this part of the proof, it is enough to verify that $tgt(\psi) = \lim_{i \to \omega} tgt(\psi_i) = t$.

- We verify that P is a tree domain, cfr. Dfn. 5.1.6. Let $pq \in P$, then $pq \in \mathtt{Pos}(tgt(\psi_{k_{|pq|}+1}))$, implying that $p \in \mathtt{Pos}(tgt(\psi_{k_{|pq|}+1}))$, and therefore that $p \in \mathtt{Pos}(tgt(\psi_{k_{|p|+1}}))$. Hence, $p \in P$. Let $pj \in P$ and i such that $1 \leqslant i \leqslant j$. Observing $|pj| = |pi|$, a straightforward argument based on $\psi_{k_{|pj|}+1}$ yields $pi \in P$.
- We verify that t is a well-defined term, cfr. Dfn. 5.1.7. Let $p \in P$, $f/m := F(p)$, and $i < \omega$. Observe $f = \psi_{k_{|p|}+1}(p) = \psi_{k_{|p|+1}+1}(p)$. Then $pi \in P$ iff $pi \in \mathtt{Pos}(\psi_{k_{|pi|}+1})$ iff $i \leqslant m$.
- We verify that $t = \lim_{i \to \omega} tgt(\psi_i)$. Let $n < \omega$, $j > k_n$, and p a position verifying $|p| \leqslant n$, so that $k_{|p|} \leqslant k_n$, implying in turn $k_{|p|} + 1 \leqslant j$. Then $p \in \mathtt{Pos}(t)$ iff $p \in \mathtt{Pos}(tgt(\psi_{k_{|p|}+1}))$ iff $p \in \mathtt{Pos}(tgt(\psi_j))$, and in such case, $t(p) = tgt(\psi_{k_{|p|}+1})(p) = tgt(\psi_j)(p)$. Hence $\mathtt{dist}(tgt(\psi_j), t) < 2^{-n}$. Consequently, $t = \lim_{i \to \omega} tgt(\psi_i)$.

We prove item (b). For all $i < \omega$, $mind(\psi_i) \geqslant mind(\psi) > n$, so that an easy induction on i using an argument similar to that just described for binary composition yields $\mathtt{dist}(src(\psi), tgt(\psi_i)) < 2^{-n}$. Recall that $tgt(\psi) = \lim_{i \to \omega} tgt(\psi_i)$, then there exists some k such that $\mathtt{dist}(tgt(\psi_j), tgt(\psi)) < 2^{-n}$ if $j > k$. Then $\mathtt{dist}(src(\psi), tgt(\psi_{k+1})) < 2^{-n}$ and $\mathtt{dist}(tgt(\psi_{k+1}), tgt(\psi)) < 2^{-n}$. We conclude by Lemma 5.1.25.

Assume $\psi = f(\psi_1, \ldots, \psi_m)$ and that it is not an infinitary multistep. Then ψ being convergent implies that all ψ_i are. Therefore a straightforward argument based on IH implies item (a) to hold. Moreover, the way in which *src*, *tgt* and *mind* for this case, implies that a natural inductive argument yields also item (b).

Assume $\psi = \mu(\psi_1, \ldots, \psi_m)$, and that it is not an infinitary multistep. Then ψ being convergent implies that ψ_i is if x_i occurs in the right-hand side of μ, thus IH:(a) implies that $tgt(\psi_i)$ is defined for those ψ_i. Hence, definition of *tgt* for this case yields item (a).

[11] A possible shortcut from here is observing that the sequence $\langle tgt(\psi_i) \rangle_{i<\omega}$ is Cauchy-convergent, and therefore has a limit. We can refer to Thm. 12.2.1 in [BKdV03], or its proof.

On the other hand, $mind(\psi) = 0$ contradicting the hypotheses of item (b). Thus we conclude. □

Lemma 5.2.18. *Let C be a context in $Ter(\Sigma)$ having k holes, and $\psi_1, \ldots, \psi_k$ proof terms. Then $mind(C[\psi_1, \ldots, \psi_k]) = min\{mind(\psi_i) + |\mathtt{Bpos}(C, i)| \ / \ 1 \leqslant i \leqslant k\}$.*

Proof. An easy, although somewhat cumbersome, induction on $max\{|\mathtt{Bpos}(C, i)|\}$ suffices. If $C = \Box$, then both sides of the equation in the lemma conclusion equates to ψ, thus we conclude.

Assume $C = f(C_1, \ldots, C_m)$.
Observe that $C[\psi_1, \ldots, \psi_k] = f(C_1[\psi_{1_1}, \ldots, \psi_{1_{q1}}], \ldots, C_m[\psi_{m_1}, \ldots, \psi_{m_{qm}}])$, where $\{\psi_{j_i}\} = \{\psi_1, \ldots, \psi_k\}$. Consequently, for any i such that $1 \leqslant i \leqslant k$, $\mathtt{Bpos}(C, i) = e\,p$ for some e verifying $1 \leqslant e \leqslant m$, and therefore $p = \mathtt{Bpos}(C_e, l)$ for some l. In turn, this implies $|\mathtt{Bpos}(C, i)| = 1 + |\mathtt{Bpos}(C_e, l)|$. Conversely, for any e such that $1 \leqslant e \leqslant m$, and for any $\mathtt{Bpos}(C_e, i)$, there is an index j such that $\mathtt{Bpos}(C, j) = e \cdot \mathtt{Bpos}(C_e, i)$. Furthermore, $mind(C[\psi_1, \ldots, \psi_k]) = 1 + min\{mind(C_j[\psi_{j_1}, \ldots, \psi_{j_{qj}}]) \ / \ 1 \leqslant j \leqslant m\}$.

Let j minimal for $mind(\psi_j) + |\mathtt{Bpos}(C, j)|$, so that showing $mind(C[\psi_1, \ldots, \psi_k]) = mind(\psi_j) + |\mathtt{Bpos}(C, j)|$ is enough to conclude. Let e, i such that $\mathtt{Bpos}(C, j) = e \cdot \mathtt{Bpos}(C_e, i)$. The existence of some j', i' such $\mathtt{Bpos}(C, j') = e \cdot \mathtt{Bpos}(C_e, i')$ and $mind(\psi'_j) + |\mathtt{Bpos}(C_e, i')| < mind(\psi_j) + |\mathtt{Bpos}(C_e, i)|$ would contradict minimality of ψ_j w.r.t. C, so that j, i are minimal for $mind(\psi_j) + |\mathtt{Bpos}(C_e, i)|$. Therefore, applying IH on C_j, yields that $mind(C_e[\psi_{e_1}, \ldots, \psi_{e_{qe}}]) = mind(\psi_j) + |\mathtt{Bpos}(C_e, i)|$.

Assume for contradiction the existence of some m, h such that
$mind(C_h[\psi_{h_1}, \ldots, \psi_{h_{qh}}]) < mind(C_e[\psi_{e_1}, \ldots, \psi_{h_{eh}}])$. Applying IH on C_h we obtain $mind(C_h[\psi_{h_1}, \ldots, \psi_{h_{qh}}]) = mind(\psi_g) + |\mathtt{Bpos}(C_h, f)|$ for some f and g such that $\mathtt{Bpos}(C, g) = h \cdot \mathtt{Bpos}(C_h, f)$. But then our assumption would imply $mind(\psi_g) + |\mathtt{Bpos}(C, g)| = mind(\psi_g) + |\mathtt{Bpos}(C_h, f)| + 1 < mind(\psi_j) + |\mathtt{Bpos}(C_e, i)| + 1 = mind(\psi_j) + |\mathtt{Bpos}(C, j)|$, contradicting minimality of j w.r.t. C.

Hence, $mind(C[\psi_1, \ldots, \psi_k]) = 1 + mind(C_e[\psi_{e_1}, \ldots, \psi_{e_{qe}}]) = mind(\psi_j) + |\mathtt{Bpos}(C, j)|$. Thus we conclude. □

Some properties related with convergence follow.

Lemma 5.2.19. *Let $\psi = f(\psi_1, \ldots, \psi_m)$ be a convergent infinitary multistep, and i such that $1 \leqslant i \leqslant m$. Then ψ_i is a convergent infinitary multistep.*

Proof. Dfn. 5.2.2 yields immediately that ψ_i is an infinitary multistep. Moreover, $f(\psi_1, \ldots, \psi_m)$ being convergent means the existence of a convergent tgt_T-reduction sequence δ such that $f(\psi_1, \ldots, \psi_m) \xrightarrow[tgt_T]{\delta} t$ and t is a tgt_T-normal form, i.e. $t \in Ter^{\infty}(\Sigma)$. Observe that $mind(\delta) > 0$, since f does not occur in any left-hand side of a rule in tgt_T. Then Lem. 5.1.45 implies $t = f(t_1, \ldots, t_m)$. In turn, Lem. 5.1.48 implies $\psi_i \xrightarrow[tgt_T]{\delta|_i} t_i$. Thus we conclude. □

Lemma 5.2.20. *Let $\psi = f(\psi_1, \ldots, \psi_m)$ be a proof term. Then ψ is convergent iff ψ_i is convergent for all i such that $1 \leqslant i \leqslant m$.*

Proof. If ψ is an infinitary multistep, then the $\Rightarrow$) direction is an immediate corollary of Lem. 5.2.19. For the $\Leftarrow$) direction, recall that for any i, ψ_i being convergent means the existence of a tgt_T-reduction sequence δ_i verifying $\psi_i \xrightarrow[tgt_T]{\delta_i} t_i$ where $t_i \in Ter^\infty(\Sigma)$. Then $f(\psi_1, \ldots, \psi_m) \xrightarrow[tgt_T]{\delta} f(t_1, \ldots, t_m)$, where $\delta := (1 \cdot \delta_1); \ldots; (m \cdot \delta_m)$, and $i \cdot \delta_i$ is defined as follows: $\mathsf{length}(i \cdot \delta_i) := \mathsf{length}(\delta_i)$ and $i \cdot \delta_i[\alpha] := \langle f(t_1, \ldots, \phi, \ldots \psi_m), ip, \mu \rangle$ where $\delta_i[\alpha] = \langle \phi, p, \mu \rangle$. A simple transfinite induction yields $f(t_1, \ldots, t_{i-1}, \psi_i, \psi_{i+1}, \ldots, \psi_m) \xrightarrow[tgt_T]{i \cdot \delta_i} f(t_1, \ldots, t_{i-1}, t_i, \psi_{i+1}, \ldots, \psi_m)$.

If ψ is not an infinitary multistep, then the result is an immediate consequence of Dfn. 5.2.8, case (4). Thus we conclude. □

Lemma 5.2.21. *Let C be a context in $Ter^\infty(\Sigma)$ having exactly m holes, and $\psi_1, \ldots, \psi_m$ proof terms. Then $C[\psi_1, \ldots, \psi_m]$ is convergent iff ψ_i is convergent for all suitable i.*

Proof. A straightforward induction on $max\{|\mathsf{Bpos}(C, i)| \ / \ 1 \leqslant i \leqslant m\}$, resorting to Lem. 5.2.20 in the inductive case, suffices to conclude. □

Lemma 5.2.22. *Let $\mu : l[x_1, \ldots, x_m] \to h[x_1, \ldots, x_m]$ be a rule included in a certain TRS; and $\psi_1, \ldots, \psi_m$ proof terms. Then $\psi = \mu(\psi_1, \ldots, \psi_m)$ is convergent iff ψ_i is convergent for all i such that x_i occurs in $h[x_1, \ldots, x_m]$.*

Proof. Assume that ψ is an infinitary multistep. We verify $\Rightarrow$). Convergence of ψ implies $\psi \xrightarrow[tgt_T]{\delta} t$ for some reduction sequence δ, where $t \in Ter^\infty(\Sigma)$. Notice that $mind(\delta) > 0$ would imply $t(\epsilon) = \mu$ (cfr. Lem. 5.1.45), contradicting $t \in Ter^\infty(\Sigma)$. Therefore $mind(\delta) = 0$, implying $\delta = \delta_1; \langle \chi, \epsilon, \underline{\nu} \rangle, \delta_2$ where $mind(\delta_1) > 0$. In turn, $mind(\delta_1) > 0$ implies that $tgt(\delta_1) = \chi = \mu(\chi_1, \ldots, \chi_m)$ where $\psi_i \xrightarrow[tgt_T]{\delta\,|_i} \chi_i$, cfr. Lem. 5.1.45 and Lem 5.1.48. Hence $\underline{\nu} = \underline{\mu} : \mu(x_1, \ldots, x_m) \to h[x_1, \ldots, x_m]$, implying $src(\delta_2) = h[\chi_1, \ldots, \chi_m]$. Observe that χ_i occurs in $src(\delta_2)$ iff x_i occurs in h. We analyse two cases:

- $h[x_1, \ldots, x_m] = x_j$, so that $src(\delta_2) = \chi_j$. In this case $\psi_j \xrightarrow{\delta\,|_j} \chi_j \xrightarrow{\delta_2} t$. We conclude by observing that only convergence of ψ_j is required in this case.
- $h \notin \mathtt{Var}$. In this case $h[\chi_1, \ldots, \chi_m] \xrightarrow{\delta_2} t$. Observe that all the steps in δ_2 lies "below" (an argument of) h. Then Lem. 5.1.49 implies $t = h[t_1, \ldots, t_m]$ and, moreover, that a reduction sequence δ'_i exists which verifies $\chi_i \xrightarrow{\delta'_i} t_i$ for all i such that x_i occurs in $h[x_1, \ldots x_m]$. Therefore, for any of those indices, say i, $\psi_i \xrightarrow{\delta_1\,|_i} \chi_i \xrightarrow{\delta'_i} t_i$. Thus we conclude.

To verify the $\Leftarrow$) direction, observe that all the ψ_i corresponding to variables occurring in h being convergent implies $\psi \to h[\psi_1, \ldots \psi_m] \xrightarrow{\delta_1} h[t_1, \ldots, \psi_m] \ldots \xrightarrow{\delta_m} h[t_1, \ldots, t_m]$, where eventually some δ_i are performed more than once, if the corresponding x_i occurs more than once in $h[x_1, \ldots, x_m]$. Hence ψ is tgt_T-WN^∞, i.e. it is a convergent infinitary multistep.

Finally, if ψ is not an infinitary multistep, then Dfn. 5.2.8, case (4), allows to conclude immediately. □

5.3 Permutation equivalence

In this section, we present a characterisation of the *equivalence of reductions*, more precisely of contraction activities, for *convergent* infinitary rewriting. A study of equivalence comprising non-convergent, as well as convergent, reductions, is left for further investigation.

As described for the finitary case in Section 2.2.3, equivalence is formally characterised by defining an equivalence relation on the set of proof terms. In fact, the definition we present in the following extends its finitary counterpart, namely Dfn. 2.2.8, preserving its basic features. Equivalence of reductions is formalised by resorting to the notion of *permutation equivalence*. Moreover, the definition of the permutation equivalence relation on the set of infinitary proof terms is based on equational logic, similarly to Dfn. 2.2.8, and additionally, the set of basic schemas for infinitary permutation equivalence is the result of adding an additional schema to those presented in the finitary definition.

We remark that this characterisation of the equivalence of infinitary reductions involves *equational logic* to be performed *on infinitary objects*, namely the proof terms. On the other hand, the mere extension of Dfn. 2.2.8, as it is presented, in the sense of allowing infinitary proof terms to be included in equational judgements, does not suffice to obtain an adequate characterisation of infinitary permutation equivalence. There are several challenges, related specifically to infinite concatenation, which must be addressed. We discuss these challenging issues by presenting some examples in the following.

5.3.1 Motivating examples

Consider the rules $\mu : f(x) \to g(x)$ and $\rho : m(x) \to j(x)$, and the reduction sequences:

$$m(f^\omega) \xrightarrow{\mu} m(g(f^\omega)) \xrightarrow{\mu} m(g^2(f^\omega)) \xrightarrow{\mu}\!\!\!\!\twoheadrightarrow m(g^\omega) \xrightarrow{\rho} j(g^\omega) \tag{5.6}$$

$$m(f^\omega) \xrightarrow{\rho} j(f^\omega) \xrightarrow{\mu} j(g(f^\omega)) \xrightarrow{\mu} j(g^2(f^\omega)) \xrightarrow{\mu}\!\!\!\!\twoheadrightarrow j(g^\omega)$$

where we annotate the arrows with the rule used in each step or sequence. These reduction sequences involve exactly the same steps, namely a μ step for each occurrence of f in $m(f^\omega)$, plus a ρ step for the external m. Therefore, we should be able to consider these reduction sequences as equivalent. Independently of the representation of these reduction sequences by proof terms, let us try to apply the notion of permutation equivalence to justify our assertion about their equivalence. To this effect, we should obtain either of these reduction sequences from the other, as the result of a series of permutations of contiguous steps.

The problem here is that the ρ step must be permuted with an *infinite* number of μ steps. If we try to transform the former equation into the latter one, there is not even a *first* definite μ step with which the ρ step can be permuted. If we go the opposite direction, for any $n < \omega$, the leading ρ step can be permuted with the leading n μ steps, resulting in

$$m(f^\omega) \xrightarrow{\mu} m(g(f^\omega)) \xrightarrow{\mu} \ldots \xrightarrow{\mu} m(g^n(f^\omega)) \xrightarrow{\rho} j(g^n(f^\omega)) \xrightarrow{\mu}\!\!\!\!\twoheadrightarrow j(g^\omega)$$

In any case, an infinite number of μ steps remain past the ρ step, preventing us to conclude the equivalence of the original reduction sequences.

This situation is reflected if we try to justify the equivalence of the proof terms denoting the given reduction sequences, namely

$$(\cdot_{i<\omega}\ m(g^i(\mu(f^\omega)))) \cdot \rho(g^\omega) \quad \text{and} \quad \rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega)))) \tag{5.7}$$

by means of Dfn. 2.2.8. On one hand, there is no "last" element in the infinite concatenation $\cdot_{i<\omega}\ m(g^i(\mu(f^\omega)))$, having the form $m(\ldots)$, which could be joined, by applying a number of times the (Assoc) schema, with $\rho(g^\omega)$, to subsequently apply the (InOut) schema. Cfr. the second example of finitary permutation equivalence judgement given in Section 2.2.3, where this idea is used to permute the last step of a finitary sequential proof term with the preceding one. On the other hand, for any $n < \omega$, the ρ step can be postponed after the n leading μ steps. The following remark is useful here, as well as in later examples.

Remark 5.3.1. *We recall from Dfn. 5.2.8 that $\cdot_{i<\omega}\ \psi_i$ is just a concise notation for $\psi_0 \cdot (\psi_1 \cdot (\psi_2 \cdot \ldots))$. Therefore, $\psi_0 \cdot (\cdot_{i<\omega}\ \psi_{1+i})$ is a different concise notation for* the same *proof term.*

Having this observation in mind, consider this permutation equivalence judgement:

$$\begin{aligned}
&\rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega)))) \\
&\quad\approx\ (\rho(f^\omega) \cdot j(\mu(f^\omega))) \cdot (\cdot_{i<\omega}\ j(g^{1+i}(\mu(f^\omega)))) \\
&\quad\approx\ \rho(\mu(f^\omega)) \cdot (\cdot_{i<\omega}\ j(g^{1+i}(\mu(f^\omega)))) \\
&\quad\approx\ (m(\mu(f^\omega)) \cdot \rho(g(f^\omega))) \cdot (\cdot_{i<\omega}\ j(g^{1+i}(\mu(f^\omega)))) \\
&\quad\approx\ m(\mu(f^\omega)) \cdot (\rho(g(f^\omega)) \cdot j(g(\mu(f^\omega)))) \cdot (\cdot_{i<\omega}\ j(g^{2+i}(\mu(f^\omega)))) \\
&\quad\approx\ m(\mu(f^\omega)) \cdot \rho(g(\mu(f^\omega))) \cdot (\cdot_{i<\omega}\ j(g^{2+i}(\mu(f^\omega)))) \\
&\quad\approx\ m(\mu(f^\omega)) \cdot (m(g(\mu(f^\omega))) \cdot \rho(g^2(f^\omega))) \cdot (\cdot_{i<\omega}\ j(g^{2+i}(\mu(f^\omega)))) \\
&\quad\approx\ m(\mu(f^\omega)) \cdot \Big(m(g(\mu(f^\omega))) \cdot \big(\rho(g^2(f^\omega)) \cdot j(g^2(\mu(f^\omega)))\big)\Big) \cdot \\
&\qquad\quad (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega)))) \\
&\quad\ \vdots \\
&\quad\approx\ m(\mu(f^\omega)) \cdot m(g(\mu(f^\omega))) \cdot m(g^2(\mu(f^\omega))) \cdot \ldots \cdot m(g^{n-1}(\mu(f^\omega))) \cdot \\
&\qquad\quad \rho(g^n(f^\omega)) \cdot (\cdot_{i<\omega}\ j(g^{n+i}(\mu(f^\omega))))
\end{aligned}$$

which repeats a pattern formed by the application of (Assoc) to join the ρ step with the following μ step, then (OutIn) to obtain a simultaneous contraction of these steps, and subsequently (InOut) to get their sequential contraction where the μ step precedes the ρ step. This pattern is similar to that described in the second example in Section 2.2.3, with (OutIn) and (InOut) applied in reverse order. Note that disregarding the value of n, an infinite concatenation comes after the ρ step. Hence, there is no way to justify, as desired, the equivalence of the two given proof terms, using just the finitary definition on infinitary proof terms.

Moreover, observe that all the steps involved in the reduction sequences in (5.6) can be performed *simultaneously* from the term $m(f^\omega)$; the infinitary multistep $\rho(\mu^\omega)$ denotes such infinite simultaneous contraction. The involved steps can be contracted in many other different forms, such as those denoted by the proof term $\rho(\cdot_{i<\omega}\ g^i(\mu(f^\omega)))$ or $\rho(f^\omega) \cdot j(\mu^\omega)$. A sound characterisation of permutation equivalence for infinitary rewriting must state that all these proof terms are equivalent among themselves, and also to any of the sequential proof terms given in (5.7). Let us take

$$\rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega)))) \quad \text{and} \quad \rho(\mu^\omega) \tag{5.8}$$

To prove the equivalence between these proof terms, all the μ steps must be "packed" (cfr. Section 1.3.2 and Section 2.2.3) from $\cdot_{i<\omega}\ j(g^i(\mu(f^\omega)))$ into an infinitary multistep, or conversely, "unpacked" from $j(\mu^\omega)$ to form an infinite concatenation. Again, Dfn. 2.2.8 allows to obtain the desired result only for a finite number of μ steps. Consider e.g. the following permutation equivalence judgement.

$$\begin{aligned}
&\rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega))))\\
&\quad\approx \Big(\rho(f^\omega) \cdot \big(j(\mu(f^\omega)) \cdot (j(g(\mu(f^\omega))) \cdot j(g(g(\mu(f^\omega)))))\big)\Big) \cdot (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega)))) && (5.9)\\
&\quad\approx \Big(\rho(f^\omega) \cdot \big(j(\mu(f^\omega)) \cdot j(g(\mu(f^\omega)) \cdot g(g(\mu(f^\omega))))\big)\Big) \cdot (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega))))\\
&\quad\approx \Big(\rho(f^\omega) \cdot \big(j(\mu(f^\omega)) \cdot j(g(\mu(f^\omega) \cdot g(\mu(f^\omega))))\big)\Big) \cdot (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega))))\\
&\quad\approx \Big(\rho(f^\omega) \cdot \big(j(\mu(f^\omega)) \cdot j(g(\mu^2(f^\omega)))\big)\Big) \cdot (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega)))) && (5.10)\\
&\quad\approx \big(\rho(f^\omega) \cdot j(\mu(f^\omega) \cdot g(\mu^2(f^\omega)))\big) \cdot (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega))))\\
&\quad\approx (\rho(f^\omega) \cdot j(\mu^3(f^\omega))) \cdot (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega)))) && (5.11)\\
&\quad\approx \rho(\mu^3(f^\omega)) \cdot (\cdot_{i<\omega}\ j(g^{3+i}(\mu(f^\omega))))
\end{aligned}$$

In this judgement, many applications of (Assoc) yield (5.9), preparing the structure where the successive "packing" of steps can take place. (Struct) is applied twice from $j(g(\mu(f^\omega))) \cdot j(g(g(\mu(f^\omega))))$, this is needed to apply (OutIn) resulting in (5.10). (Struct) is again followed by (OutIn) to obtain (5.11), and a last application of (OutIn) yields the final result. By repeating the "(Struct)-and-then-(OutIn)" pattern, a judgement can be built to justify

$$\rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega)))) \quad\approx\quad \rho(\mu^n(f^\omega)) \cdot (\cdot_{i<\omega}\ j(g^{n+i}(\mu(f^\omega))))$$

for any $n < \omega$, so that an infinite concatenation comes after the leading infinitary multistep including the ρ step and the n external μ steps. Therefore, just applying Dfn. 2.2.8, the equivalence of the proof terms in (5.8) cannot be attained.

Let us analyse a different case, using the rules $\mu : f(x) \to g(x)$ and $\nu : g(x) \to k(x)$. Consider the reduction sequences

$$f^\omega \to g(f^\omega) \to g^2(f^\omega) \twoheadrightarrow g^\omega \to k(g^\omega) \to k^2(g^\omega) \twoheadrightarrow k^\omega \tag{5.12}$$

$$f^\omega \to g(f^\omega) \to k(f^\omega) \to k(g(f^\omega)) \to k^2(f^\omega) \twoheadrightarrow k^\omega \tag{5.13}$$

These reduction sequences comprise the contraction of exactly the same steps: a μ step for any occurrence of f in f^ω, plus a ν step for any of the created occurrences of g. They are different only by the order in which these steps are contracted. Therefore, we should be able to conclude that the proof terms

$$\begin{aligned}
&(\cdot_{i<\omega}\ g^i(\mu(f^\omega))) \cdot (\cdot_{i<\omega}\ k^i(\nu(g^\omega))) && (5.14)\\
&\cdot_{i<\omega}\ (k^i(\mu(f^\omega)) \cdot k^i(\nu(g^\omega)))
\end{aligned}$$

which denote respectively the just introduced reduction sequences, are equivalent.

As in the previous examples, the permutation equivalence of these proof terms cannot be justified by resorting to Dfn. 2.2.8. The situation is even more complex in this case, than in the example given by (5.6) and (5.7). To transform (5.12) into (5.13), we should permute an *infinite* number of ν steps; in turn, each of these ν steps must be permuted with an *infinite* number of μ steps. That is, infinity is involved in the permutation equivalence reasoning of this case, in two different dimensions. The need to cope with this phenomenon has a great influence on the design of the formal definition of the permutation equivalence relation for infinitary proof terms.

5.3.2 The formal definition

In order to obtain an adequate characterisation of infinitary permutation equivalence, we add three elements to Dfn. 2.2.8.

Two of these added elements are a basic equation schema and a contextual equational logic rule, which allow to extend to *infinite concatenations*, respectively, the (Struct) scheme and the closure by operations, which the finitary permutation equivalence definition provide for binary concatenation only. Namely, the characterisation of infinitary permutation equivalence includes the equation schema:

$$(\mathsf{InfStruct}) \quad \cdot_{i<\omega}\ f(\psi_i^1, \ldots, \psi_i^m) \quad \sim \quad f(\cdot_{i<\omega}\ \psi_i^1, \ldots, \cdot_{i<\omega}\ \psi_i^m)$$

and the equational rule:

$$\frac{\psi_i \sim \phi_i \quad \text{for all } i < \omega}{\cdot_{i<\omega}\ \psi_i \ \approx\ \cdot_{i<\omega}\ \phi_i}\ \mathsf{InfComp}$$

The third added element is an equational rule which allows to incorporate the notion of *limit* into the permutation equivalence judgements. The form of this rule is:

$$\frac{\begin{array}{l} \text{for all } k < \omega \\ \text{exists } \chi_k, \psi'_k, \phi'_k \end{array} \quad \left\{ \begin{array}{ll} \psi \approx_B \chi_k \cdot \psi'_k & mind(\psi'_k) > k \\ \phi \approx_B \chi_k \cdot \phi'_k & mind(\phi'_k) > k \end{array} \right.}{\psi \ \approx \ \phi}\ \mathsf{Lim}$$

where $\approx_B$ denotes a restriction of the $\approx$ relation, which we discuss later on. The Lim rule can be described as follows: if ψ and ϕ can be proven permutation equivalent, respectively, to two arbitrarily "similar" proof terms, then we can conclude that ψ and ϕ are, themselves, permutation equivalent.

The notion of "similarity" between proof terms we use in this rule, reflect the similarity of the denoted contraction activities. It is based on two elements: the concatenation symbol, i.e. the dot, and the notion of minimum activity depth. By resorting to the dot, we can separate a prefix of any contraction activity, not only of reduction sequences. In any proof term having the form $\xi \cdot \theta$, the activity denoted by ξ precedes that denoted by θ. The use of the dot allows to define the degree of "similarity" between two proof terms, in relation to what follows a common prefix. Namely, given two proof terms $\xi \cdot \theta_1$ and $\xi \cdot \theta_2$, the less significant θ_1 and θ_2 are, the more "similar" we consider $\xi \cdot \theta_1$ and $\xi \cdot \theta_2$. In turn, the "significance" of a proof term is considered to be the inverse of its minimum activity depth. A proof term ξ verifying $mind(\xi) = 0$ includes root activity; we consider that such proof terms have the greatest significance. A greater value of $mind(\xi)$ indicates a greater context which is not affected by (the activity denoted by) ξ, and hence a smaller significance.

We want to remark another aspect related to the Lim rule: we consider that the "stacking" of uses of this rule, i.e. to use Lim in the derivation leading to a premise of another Lim application, should not be necessary to obtain an adequate characterisation of infinitary permutation equivalence. Therefore, we resort to a separate relation $\approx_B$, which we call *base permutation equivalence*, and which is the closure of the equation schema instances by all the rules except for Lim, in the premises of the Lim rule.

We note that the added rules InfComp and Lim have an infinite number of premises. In order to obtain a proper definition, and also to have a way to reason about the permutation equivalence relation by transfinite induction, we organise the following definition in *layers*, similarly as the definition of the set of infinitary proof terms in Section 5.2, cfr. Dfn. 5.2.8 and Dfn. 5.2.9.

The formal definition of the permutation equivalence relation on the set of infinitary proof terms follows.

Definition 5.3.2 (Layer of base permutation equivalence)**.** *Let α be a countable ordinal. We define the α-th* level of base permutation equivalence*, notation $\overset{\alpha}{\approx}_B$, as follows: given ψ and ϕ proof terms, $\psi \overset{\alpha}{\approx}_B \phi$ iff the equation $\psi \overset{\alpha}{\sim} \phi$ can be obtained by means of the equational logic system whose basic equations are the valid instances of the following schemas:*

$$
\begin{array}{llcl}
(\mathsf{IdLeft}) & src(\psi) \cdot \psi & \sim & \psi \\
(\mathsf{IdRight}) & \psi \cdot tgt(\psi) & \sim & \psi \\
(\mathsf{Assoc}) & \psi \cdot (\phi \cdot \chi) & \sim & (\psi \cdot \phi) \cdot \chi \\
(\mathsf{Struct}) & f(\psi_1, \ldots, \psi_m) \cdot f(\phi_1, \ldots, \phi_m) & \sim & f(\psi_1 \cdot \phi_1, \ldots, \psi_m \cdot \phi_m) \\
(\mathsf{InfStruct}) & \cdot_{i<\omega}\, f(\psi_i^1, \ldots, \psi_i^m) & \sim & f(\cdot_{i<\omega}\, \psi_i^1, \ldots, \cdot_{i<\omega}\, \psi_i^m) \\
(\mathsf{OutIn}) & \mu(\psi_1, \ldots, \psi_m) & \sim & \mu(s_1, \ldots, s_m) \cdot r[\psi_1, \ldots, \psi_m] \\
(\mathsf{InOut}) & \mu(\psi_1, \ldots, \psi_m) & \sim & l[\psi_1, \ldots, \psi_m] \cdot \mu(t_1, \ldots, t_m)
\end{array}
$$

where $\mu : l \to r$, $s_i = src(\psi_i)$, $t_i = tgt(\psi_i)$, and an instance of an equation is valid iff both the lhs and rhs are convergent proof terms, cfr. Dfn. 5.2.9. For (InOut)*, notice that the target of each of the ψ_i must be defined, since all the t_i occur in the right-hand side of the equation schema; therefore, all the ψ_i must be convergent proof terms for an instance of this schema to be valid.*

Equational logic rules are defined by transfinite recursion on α as follows

$$
\frac{}{\psi \overset{1}{\sim} \psi}\ \mathsf{Refl}
\qquad
\frac{\psi \sim \phi \text{ is a basic equation}}{\psi \overset{1}{\sim} \phi}\ \mathsf{Eqn}
$$

$$
\frac{\psi \overset{\alpha_1}{\sim} \phi}{\phi \overset{\alpha_1+1}{\sim} \psi}\ \mathsf{Symm}
\qquad
\frac{\psi \overset{\alpha_1}{\sim} \phi \quad \phi \overset{\alpha_2}{\sim} \xi}{\psi \overset{\alpha_1+\alpha_2+1}{\sim} \xi}\ \mathsf{Trans}
$$

$$
\frac{\psi_1 \overset{\alpha_1}{\sim} \phi_1 \quad \ldots \quad \psi_n \overset{\alpha_n}{\sim} \phi_n \quad f/n \in \Sigma}{f(\psi_1, \ldots, \psi_n) \overset{\alpha_1+\ldots+\alpha_n+1}{\sim} f(\phi_1, \ldots, \phi_n)}\ \mathsf{Fun}
$$

$$
\frac{\psi_1 \overset{\alpha_1}{\sim} \phi_1 \quad \ldots \quad \psi_n \overset{\alpha_n}{\sim} \phi_n \quad \mu/n \textit{ is a rule symbol}}{\mu(\psi_1, \ldots, \psi_n) \overset{\alpha_1+\ldots+\alpha_n+1}{\sim} \mu(\phi_1, \ldots, \phi_n)}\ \mathsf{Rule}
$$

$$
\frac{\psi_1 \overset{\alpha_1}{\sim} \phi_1 \quad \psi_2 \overset{\alpha_2}{\sim} \phi_2}{\psi_1 \cdot \psi_2 \overset{\alpha_1+\alpha_2+1}{\sim} \phi_1 \cdot \phi_2}\ \mathsf{Comp}
\qquad
\frac{\psi_i \overset{\alpha_i}{\sim} \phi_i \quad \text{for all } i < \omega}{\cdot_{i<\omega}\, \psi_i \overset{\Sigma_{i<\omega}\alpha_i}{\sim} \cdot_{i<\omega}\, \phi_i}\ \mathsf{InfComp}
$$

Definition 5.3.3 (Base permutation equivalence)**.** *Let ψ, ϕ be proof terms. We say that ψ and ϕ are* base-permutation equivalent*, notation $\psi \approx_B \phi$, iff $\psi \overset{\alpha}{\approx}_B \phi$ for some $\alpha < \omega_1$.*

Definition 5.3.4 (Layer of permutation equivalence)**.** *Let α be a countable ordinal. We define the α-th* level of permutation equivalence*, notation $\overset{\alpha}{\approx}$, as follows: given ψ and ϕ proof terms, $\psi \overset{\alpha}{\approx} \phi$ iff the equation $\psi \overset{\alpha}{\sim} \phi$ can be obtained by means of the equational logic system whose basic equations are those described in Dfn. 5.3.2, and the set of equational logic rules is the result of adding the rule* Lim *defined as follows*

$$\frac{\begin{array}{ll}\text{for all } k < \omega & \left\{\begin{array}{ll}\psi \overset{\alpha_k}{\approx}_B \chi_k \cdot \psi'_k & mind(\psi'_k) > k \\ \phi \overset{\beta_k}{\approx}_B \chi_k \cdot \phi'_k & mind(\phi'_k) > k\end{array}\right. \\ \text{exists } \chi_k, \psi'_k, \phi'_k & \end{array}}{\psi \overset{\alpha}{\sim} \phi \qquad \text{where } \alpha = \sum_{i<\omega} \alpha_i + \sum_{i<\omega} \beta_i} \ \mathsf{Lim}$$

to the rules introduced in Dfn. 5.3.2.

Definition 5.3.5 (Permutation equivalence)**.** *Let ψ, ϕ be proof terms. We say that ψ and ϕ are* permutation equivalent*, notation $\psi \approx \phi$, iff $\psi \overset{\alpha}{\approx} \phi$ for some $\alpha < \omega_1$.*

Observe that for any countable ordinal α, $\overset{\alpha}{\approx}_B \subseteq \overset{\alpha}{\approx}$, and therefore $\approx_B \subseteq \approx$.

As indicated prior to the definitions, the use of $\approx_B$ instead of $\approx$ in the premises of the Lim rule prevents the use of Lim in the judgements leading to the premises of a Lim application. On the other hand, this definition does allow permutation equivalence judgements including several applications of the Lim rule. E.g., the following:

$$\frac{\dfrac{\cdots \quad \begin{array}{c}\psi_1 \approx_B \xi_k \cdot \psi'_1 \\ \phi_1 \approx_B \xi_k \cdot \phi'_1\end{array} \quad \cdots}{\psi_1 \approx \phi_1} \ \mathsf{Lim} \qquad \dfrac{\cdots \quad \begin{array}{c}\psi_2 \approx_B \chi_k \cdot \psi'_2 \\ \phi_2 \approx_B \chi_k \cdot \phi'_2\end{array} \quad \cdots}{\psi_2 \approx \phi_2} \ \mathsf{Lim}}{\psi_1 \cdot \psi_2 \approx \phi_1 \cdot \phi_2} \ \mathsf{Comp}$$

is a valid permutation equivalence derivation. The Lim rule is discussed, including possible variations, in the conclusions of this thesis, cfr. Sections 6.1.3 and 6.3.

Finally, we want to point out a subtle point in Dfn. 5.3.2, regarding the notion of *valid instance* of an equation schema, particularly in relation to the schemas (OutIn) and (InOut). For an instance of either of this schemas to be valid, the corresponding instance of $\mu(\psi_1, \ldots, \psi_m)$ must be a convergent proof term. Note that this condition does not entail that all the ψ_i must be convergent as well. We recall that Dfn. 5.2.8 requires convergence only for the ψ_i occurring in $r[\psi_1, \ldots, \psi_m]$, where r is the right-hand side of μ. Let us give an example, using the rules $\mu : f(x) \to g(x)$, $\nu : g(x) \to k(x)$, $\rho : h(x, y) \to j(y)$, and $\tau : i(x) \to x$. The instances of (OutIn) and (InOut) regarding the rule ρ have this shape:

$$\begin{array}{ll}(\mathsf{OutIn}) & \rho(\psi_1, \psi_2) \sim \rho(src(\psi_1), src(\psi_2)) \cdot j(\psi_2) \\ (\mathsf{InOut}) & \rho(\psi_1, \psi_2) \sim h(\psi_1, \psi_2) \cdot \rho(tgt(\psi_1), tgt(\psi_2))\end{array}$$

The following instance of (OutIn):

$$\rho(\tau^\omega, \mu(a) \cdot \nu(a)) \sim \rho(i^\omega, f(a)) \cdot j(\mu(a) \cdot \nu(a))$$

is valid, even when the proof term standing for ψ_1, namely τ^ω, is not convergent. Observe that the *source* term of any proof term, either convergent or not, can be computed; cfr.

the comment preceding Dfn. 5.2.4. On the other hand, the *target* of τ^ω is undefined. Therefore, there is no instance of (InOut) whose left-hand side is $\rho(\tau^\omega, \mu(a) \cdot \nu(a))$, because the target of the proof term standing for ψ_1 cannot be computed, whereas it is needed to build the instance of this schema. Hence the condition about convergence of each ψ_i for instances of the (InOut) schema, detailed in Dfn. 5.3.2.

5.3.3 Some infinitary permutation equivalence judgements

It is straightforward to observe that the relation given by Dfn. 5.3.5 includes the finitary permutation equivalence relation formalised by Dfn. 2.2.8. Therefore, the examples given in Section 2.2.3 are valid infinitary permutation equivalence judgements as well.

To illustrate the application of the three elements added to obtain the characterisation of infinitary permutation equivalence, we go back to the motivation examples described in Section 5.3.1, and a variation of one of them. We use the following rules:

$$\mu : f(x) \to g(x) \qquad \nu : g(x) \to k(x) \qquad \rho : m(x) \to j(x) \qquad \tau : k(x) \to m(x)$$

As a first example, we recall the proof terms given in (5.7), page 170:

$$(\cdot_{i<\omega}\ m(g^i(\mu(f^\omega)))) \cdot \rho(g^\omega) \quad \text{and} \quad \rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega))))$$

We can prove the equivalence of these proof terms easily, by resorting to the (InfStruct) equation schema, as follows:

$$\begin{aligned}
&(\cdot_{i<\omega}\ m(g^i(\mu(f^\omega)))) \cdot \rho(g^\omega) \\
&\qquad \approx\ m(\cdot_{i<\omega}\ g^i(\mu(f^\omega))) \cdot \rho(g^\omega) \\
&\qquad \approx\ \rho(\cdot_{i<\omega}\ g^i(\mu(f^\omega))) \\
&\qquad \approx\ \rho(f^\omega) \cdot j(\cdot_{i<\omega}\ g^i(\mu(f^\omega))) \\
&\qquad \approx\ \rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega))))
\end{aligned} \tag{5.15}$$

Application of (InfStruct) allows to use (InOut) afterwards to obtain (5.15), where the infinite concatenation of the μ steps is enclosed in the ρ step. Subsequently, (OutIn) allows the reorder the contraction activity as desired, and a final application of (InfStruct) yields the expected result.

Now let us recall the proof terms presented in (5.8), namely

$$\rho(f^\omega) \cdot (\cdot_{i<\omega}\ j(g^i(\mu(f^\omega)))) \quad \text{and} \quad \rho(\mu^\omega)$$

We prove that these proof terms are permutation equivalent by resorting to the Lim rule. Consider the following derivation

$$\begin{array}{rcll}
\rho(\mu^\omega) & \approx & \rho(f^\omega) \cdot j(\mu^\omega) & \text{(OutIn)} \\
& \approx & \rho(f^\omega) \cdot j(\mu(f^\omega) \cdot g(\mu^\omega)) & \text{(OutIn)} \\
& \approx & \rho(f^\omega) \cdot (j(\mu(f^\omega)) \cdot j(g(\mu^\omega))) & \text{(Struct)} \\
& \approx & (\rho(f^\omega) \cdot j(\mu(f^\omega))) \cdot j(g(\mu^\omega)) & \text{(Assoc)} \\
& \approx & (\rho(f^\omega) \cdot j(\mu(f^\omega))) \cdot j(g(\mu(f^\omega) \cdot g(\mu^\omega))) & \text{(OutIn)} \\
& \approx & (\rho(f^\omega) \cdot j(\mu(f^\omega))) \cdot (j(g(\mu(f^\omega))) \cdot j(g^2(\mu^\omega))) & \text{(Struct)} \times 2 \\
& \approx & ((\rho(f^\omega) \cdot j(\mu(f^\omega))) \cdot j(g(\mu(f^\omega)))) \cdot j(g^2(\mu^\omega)) & \text{(Assoc)}
\end{array}$$

where the used equation schema is indicated next to each line; the legend (Struct) $\times 2$ in the next-to-last line indicates the two successive applications of that schema from $j(g(\mu(f^\omega) \cdot g(\mu^\omega)))$ needed to obtain the final result.

Let $n < \omega$. By iterating over the shown pattern of (OutIn) followed by (an ever-increasing number of applications of) (Struct) and then (Assoc), we can obtain:

$$\rho(\mu^\omega) \;\approx\; \big(\rho(f^\omega) \cdot j(\mu(f^\omega)) \cdot j(g(\mu(f^\omega))) \cdot \ldots \cdot j(g^{n-1}(\mu(f^\omega)))\big) \cdot j(g^n(\mu^\omega)) \quad (5.16)$$

where some parenthesis are omitted. On the other hand, repeated application of (Assoc) suffices to obtain

$$\begin{aligned} &\rho(f^\omega) \cdot (\cdot_{i<\omega}\, j(g^i(\mu(f^\omega)))) \\ &\quad\approx\; \big(\rho(f^\omega) \cdot j(\mu(f^\omega)) \cdot j(g(\mu(f^\omega))) \cdot \ldots \cdot j(g^{n-1}(\mu(f^\omega)))\big) \cdot (\cdot_{i<\omega}\, j(g^{n+i}(\mu(f^\omega)))) \end{aligned} \quad (5.17)$$

Moreover, $mind(j(g^n(\mu^\omega))) = mind(\cdot_{i<\omega}\, j(g^{n+i}(\mu(f^\omega)))) = n+1 > n$. For the latter, observe that for all $i < \omega$, we have $mind(j(g^{n+i}(\mu(f^\omega)))) = n+i+1 \geqslant n+1$. Note also that Lim is not used to obtain (5.16) and (5.17). Hence, Lim can be applied, where $\chi_n = \rho(f^\omega) \cdot j(\mu(f^\omega)) \cdot j(g(\mu(f^\omega))) \cdot \ldots \cdot j(g^{n-1}(\mu(f^\omega)))$, to obtain the expected result.

We go back to the last example discussed in Section 5.3.1, cfr. (5.14), involving the proof terms

$$\psi = (\cdot_{i<\omega}\, g^i(\mu(f^\omega))) \cdot (\cdot_{i<\omega}\, k^i(\nu(g^\omega))) \qquad \phi = \cdot_{i<\omega}\, (k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega)))$$

We prove that $\psi \approx \phi$, again resorting to the Lim rule. Observe the following derivation:

$$\begin{aligned} \psi &= (\cdot_{i<\omega}\, g^i(\mu(f^\omega))) \cdot (\cdot_{i<\omega}\, k^i(\nu(g^\omega))) \\ &\approx \mu(f^\omega) \cdot ((\cdot_{i<\omega}\, g(g^i(\mu(f^\omega)))) \cdot \nu(g^\omega)) \cdot (\cdot_{i<\omega}\, k(k^i(\nu(g^\omega)))) \\ &\approx \mu(f^\omega) \cdot (g(\cdot_{i<\omega}\, g^i(\mu(f^\omega))) \cdot \nu(g^\omega)) \cdot (\cdot_{i<\omega}\, k(k^i(\nu(g^\omega)))) && (5.18) \\ &\approx \mu(f^\omega) \cdot (\nu(f^\omega) \cdot k(\cdot_{i<\omega}\, g^i(\mu(f^\omega)))) \cdot k(\cdot_{i<\omega}\, k^i(\nu(g^\omega))) && (5.19) \\ &\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot k(\psi) && (5.20) \\ &\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot k(\mu(f^\omega) \cdot \nu(f^\omega) \cdot k(\psi)) \\ &\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot k(\mu(f^\omega)) \cdot k(\nu(f^\omega)) \cdot k^2(\psi) && (5.21) \end{aligned}$$

We apply (Assoc) many times, taking into account Remark 5.3.1, and then (InfStruct) on $\cdot_{i<\omega}\, g(g^i(\mu(f^\omega)))$, to obtain (5.18). The successive application of (InOut) and (OutIn) models the permutation of the first ν step with the infinite μ steps in this infinite concatenation, yielding (5.19). From this proof term, (Assoc) following by (Struct) results in (5.20). Repeating the whole argument, and then applying (Struct), leads to (5.21).

Let $n < \omega$. By iterating the just described reasoning, we can obtain

$$\psi \approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot k(\mu(f^\omega)) \cdot k(\nu(f^\omega)) \cdot \ldots \cdot k^n(\mu(f^\omega)) \cdot k^n(\nu(f^\omega)) \cdot k^{n+1}(\psi)$$

observe $mind(k^{n+1}(\psi)) = n+1$. On the other hand, repeated application of (Assoc) suffices to obtain

$$\phi \approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot k(\mu(f^\omega)) \cdot k(\nu(f^\omega)) \cdot \ldots \cdot k^n(\mu(f^\omega)) \cdot k^n(\nu(f^\omega)) \cdot \phi'$$

where $\phi' = \cdot_{i<\omega} (k^{n+1+i}(\mu(f^\omega)) \cdot k^{n+1+i}(\nu(f^\omega)))$. so that $mind(\phi') = mind(k^{n+1}(\mu(f^\omega)) \cdot k^{n+1}(\nu(f^\omega))) = n+1$. Hence we can apply Lim to obtain $\psi \approx \phi$, as expected.

We remark the use of Lim to make it possible to build this permutation equivalence argument, in which infinity appears in two different dimensions, as described in Section 5.3.1. An argument not resorting to Lim suffices to model the permutation of one ν step with an infinite number of μ steps; observe the role of (InfStruct) to this effect. In turn, the permutation of the infinite ν steps is modeled by means of a limit argument, formalised by the Lim rule.

The last example in this section shows the role of the InfComp rule in a permutation equivalence judgement. Let us consider the following proof terms

$$\psi = (\cdot_{i<\omega} (k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega)))) \cdot (\cdot_{i<\omega} m^i(\tau(k^\omega)))$$
$$\phi = \cdot_{i<\omega} (m^i(\mu(f^\omega)) \cdot m^i(\nu(f^\omega)) \cdot m^i(\tau(f^\omega)))$$

This case is similar to the previous one: in order to prove the equivalence of ψ and ϕ, an *infinite* number of τ steps must be permuted, each of them with an *infinite* number of μ and ν steps. We would like to apply an argument similar to that described for the previous example. Namely, to show that for any $n < \omega$, a prefix similar to that of ϕ can be obtained by permuting each of the first n τ steps in ψ w.r.t. an infinite number of μ steps and ν steps. In turn, the permutation of each τ step must involve the transformation of $\psi_1 = \cdot_{i<\omega} (k(k^i(\mu(f^\omega))) \cdot k(k^i(\nu(f^\omega))))$ into a proof term having the form $k(\psi')$, so that (InOut) can be applied, followed by (OutIn), as in the steps leading to (5.18) and (5.19) in the previous derivation.

But in this case, ψ_1 does not have the form $\cdot_{i<\omega} k(\xi)$, so that (InfStruct) does not apply to this proof term. On the other hand, the (Struct) equation schema can be applied *to each component* in ψ_1, as follows:

$$k(k^i(\mu(f^\omega))) \cdot k(k^i(\nu(f^\omega))) \approx k(k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega)))$$

Taking each of this one-step derivations as premises, the InfComp rule yields

$$\psi_1 \approx \cdot_{i<\omega} k(k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega)))$$

This observation enables the following judgement

$$\begin{aligned}
\psi &= (\cdot_{i<\omega} (k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega)))) \cdot (\cdot_{i<\omega} m^i(\tau(k^\omega))) \\
&\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot (\psi_1 \cdot \tau(k^\omega)) \cdot (\cdot_{i<\omega} m(m^i(\tau(k^\omega)))) && (5.22) \\
&\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot (\cdot_{i<\omega} k(k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega))) \cdot \tau(k^\omega)) \cdot (\cdot_{i<\omega} m(m^i(\tau(k^\omega)))) && (5.23) \\
&\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot (k(\cdot_{i<\omega} k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega))) \cdot \tau(k^\omega)) \cdot (\cdot_{i<\omega} m(m^i(\tau(k^\omega)))) \\
&\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot (\tau(f^\omega) \cdot m(\cdot_{i<\omega} k^i(\mu(f^\omega)) \cdot k^i(\nu(f^\omega)))) \cdot m(\cdot_{i<\omega} m^i(\tau(k^\omega))) && (5.24) \\
&\approx \mu(f^\omega) \cdot \nu(f^\omega) \cdot \tau(f^\omega) \cdot m(\psi) && (5.25)
\end{aligned}$$

This derivation is similar to that of the previous example, with the addition of the argument based on InfComp previously described. (Assoc) yields (5.22); in turn, the just referred argument leads to (5.23). (InfStruct) can be applied on this proof term, allowing in turn to permute the τ step; we obtain (5.24). Finally, (Assoc) and (Struct) lead to (5.25).

Hence, the general argument given for the previous example allows to conclude $\psi \approx \phi$, as expected.

5.3.4 Infinitary erasure

As described in Section 2.2.3, the characterisation of permutation equivalence obtained by applying equational logic on proof terms models adequately the phenomenon of *erasure* of some contraction activity by step permutation. This feature is common to the finitary and infinitary versions given in that section and Section 5.3.2 respectively.

Infinitary rewriting, and particularly the behavior of reduction sequences at limit ordinals, provoke a different form of erasure of contraction activity, which we call *infinitary erasure*. Let us describe this phenomenon by means of an example, using the rules $\kappa : f(x) \to g(f(x))$ and $\pi : a \to b$. Consider the reduction sequence

$$f(a) \xrightarrow{\pi} f(b) \xrightarrow{\kappa} g(f(b)) \xrightarrow{\kappa} g^2(f(b)) \xrightarrow{\kappa}\!\!\!\twoheadrightarrow g^\omega \tag{5.26}$$

where we decorate the arrows with the rule used in each step or sequence. We can permute the π step with each of the κ steps in turn. After the permutation of n steps we get

$$f(a) \xrightarrow{\kappa} g(f(a)) \xrightarrow{\kappa} \ldots \xrightarrow{\kappa} g^n(f(a)) \xrightarrow{\pi} g^n(f(b)) \xrightarrow{\kappa}\!\!\!\twoheadrightarrow g^\omega$$

If we resort to the notion of limit to model the permutation of the π step with *all* the κ steps, then we obtain

$$f(a) \xrightarrow{\kappa} g(f(a)) \xrightarrow{\kappa} \ldots \xrightarrow{\kappa} g^n(f(a)) \xrightarrow{\kappa} g^{n+1}(f(a)) \xrightarrow{\kappa}\!\!\!\twoheadrightarrow g^\omega \tag{5.27}$$

After the contraction of all the κ steps, there is no trace of the source of the π step. The latter is *erased* as a result of taking the limit of an infinite number of step permutations. Therefore, we consider (5.26) and (5.27) as equivalent reductions.

The characterisation of permutation equivalence for infinitary rewriting we present in this chapter models adequately the phenomenon of infinitary erasure. To verify this assertion, let us work out the just given example. The reduction sequences (5.26) and (5.27) can be denoted, respectively, by the proof terms

$$\psi = f(\pi) \cdot (\cdot_{i<\omega}\, g^i(\kappa(b))) \qquad\qquad \phi = \cdot_{i<\omega}\, g^i(\kappa(a))$$

We can prove that these terms are permutation equivalent by resorting to the Lim rule. Consider the following derivation

$$\begin{aligned}
\psi &= f(\pi) \cdot (\cdot_{i<\omega}\, g^i(\kappa(b))) \\
&\approx (f(\pi) \cdot \kappa(b)) \cdot (\cdot_{i<\omega}\, g(g^i(\kappa(b)))) \\
&\approx \kappa(a) \cdot g(f(\pi)) \cdot g(\cdot_{i<\omega}\, g^i(\kappa(b))) \\
&\approx \kappa(a) \cdot g(\psi) && (5.28) \\
&\approx \kappa(a) \cdot g(\kappa(a) \cdot g(\psi)) \\
&\approx \kappa(a) \cdot g(\kappa(a)) \cdot g^2(\psi) && (5.29)
\end{aligned}$$

The permutation of the π step with the first κ step is achieved by applying (Assoc) and then (InOut) and (OutIn). By applying (InfStruct) on $\cdot_{i<\omega}\, g(g^i(\kappa(b)))$, and then (Assoc) and (Struct), we obtain (5.28). Repeating the whole argument and then applying (Struct) yields (5.29). For any $n < \omega$, iterating over this reasoning results in

$$\psi \approx \big(\kappa(a) \cdot g(\kappa(a)) \cdot \ldots \cdot g^n(\kappa(a))\big) \cdot g^{n+1}(f(\pi)) \cdot (\cdot_{i<\omega}\, g^{n+1+i}(\kappa(b)))$$

On the other hand, a straightforward argument implies

$$\phi \approx \big(\kappa(a) \cdot g(\kappa(a)) \cdot \ldots \cdot g^n(\kappa(a))\big) \cdot (\cdot_{i<\omega}\, g^{n+1+i}(\kappa(a)))$$

so that the Lim rule can be applied to obtain $\psi \approx \phi$.

5.3.5 Basic properties of permutation equivalence

Lemma 5.3.6. *Let ψ, ϕ be convergent proof terms such that $\psi \approx \phi$. Then $src(\psi) = src(\phi)$, $tgt(\psi) = tgt(\phi)$ and $mind(\psi) = mind(\phi)$.*

Proof. We proceed by induction on α where $\psi \overset{\alpha}{\approx} \phi$, analysing the equational logic rule used in the final step of that judgement. Observe particularly that Lem 5.2.17:(a) implies both $tgt(\psi)$ and $tgt(\phi)$ to be defined. If the rule is Eqn, then we analyse the equation of which the pair $\langle\psi,\phi\rangle$ is an instance. It turns out that the only non-trivial cases are those corresponding to the (InfStruct) equation and the InfComp and Lim rules. We prove the result for each of these cases.

Assume that $\langle\psi,\phi\rangle$ is an instance of the (InfStruct) equation, i.e., that $\psi = \cdot_{i<\omega}\, f(\psi_i^1,\ldots,\psi_i^m)$ and $\phi = f(\cdot_{i<\omega}\, \psi_i^1,\ldots,\cdot_{i<\omega}\, \psi_i^m)$.

- We verify $mind(\psi) = mind(\phi)$.
  Observe that $mind(\psi) = min_{i<\omega}(mind(f(\psi_i^1,\ldots,\psi_i^m))) = mind(f(\psi_a^1,\ldots,\psi_a^m)) = 1 + min(mind(\psi_a^1),\ldots,mind(\psi_a^m)) = 1 + mind(\psi_a^b)$ where

$$\begin{aligned} mind(f(\psi_a^1,\ldots,\psi_a^m)) &\leqslant mind(f(\psi_i^1,\ldots,\psi_i^m)) \quad \text{for all } i<\omega & (5.30)\\ mind(\psi_a^b) &\leqslant mind(\psi_a^j) \quad \text{if } 1\leqslant j\leqslant m & (5.31)\end{aligned}$$

  On the other hand, $mind(\phi) = 1 + min(mind(\cdot_{i<\omega}\, \psi_i^1),\ldots,mind(\cdot_{i<\omega}\, \psi_i^m)) = 1 + mind(\cdot_{i<\omega}\, \psi_i^{b'}) = 1 + mind(\psi_{a'}^{b'})$ where

$$\begin{aligned} mind(\cdot_{i<\omega}\, \psi_i^{b'}) &\leqslant mind(\cdot_{i<\omega}\, \psi_i^j) \quad \text{if } 1\leqslant j\leqslant m & (5.32)\\ mind(\psi_{a'}^{b'}) &\leqslant mind(\psi_i^{b'}) \quad \text{for all } i<\omega & (5.33)\end{aligned}$$

  Assume for contradiction $mind(\psi_a^b) < mind(\psi_{a'}^{b'})$. Then $b \neq b'$ would imply $mind(\cdot_{i<\omega}\, \psi_i^b) \leqslant mind(\psi_a^b) < mind(\psi_{a'}^{b'}) = mind(\cdot_{i<\omega}\, \psi_i^{b'})$, contradicting (5.32), and $b = b'$ would immediately contradict (5.33). Analogously, if we assume $mind(\psi_{a'}^{b'}) < mind(\psi_a^b)$, then $a \neq a'$ would imply $mind(f(\psi_{a'}^1,\ldots,\psi_{a'}^m)) \leqslant 1 + mind(\psi_{a'}^{b'}) < 1 + mind(\psi_a^b) = mind(f(\psi_a^1,\ldots,\psi_a^m))$, contradicting (5.30), and $a = a'$ would immediately contradict (5.31). Hence we conclude.

- To verify the condition about source terms, it is enough to observe that $src(\psi) = src(\phi) = f(src(\psi_0^1),\ldots,src(\psi_0^m))$.

- We verify $tgt(\psi) = tgt(\phi)$. Observe that $tgt(\psi) = \lim_{i \to \omega} f(tgt(\psi_i^1), \ldots, tgt(\psi_i^m))$ and $tgt(\phi) = f(\lim_{i \to \omega} tgt(\psi_i^1), \ldots, \lim_{i \to \omega} tgt(\psi_i^m))$.
  Let $t_j := \lim_{i \to \omega} tgt(\psi_i^j)$, so that $tgt(\phi) = f(t_1, \ldots, t_m)$. Then it is enough to prove that $\mathtt{dist}(tgt(\psi), f(t_1, \ldots, t_m)) = 0$.
  Let $n < \omega$. Let k such that for all j, $i > k$ implies $\mathtt{dist}(tgt(\psi_i^j), t_j) < 2^{-(n-1)}$ and also $\mathtt{dist}(f(tgt(\psi_i^1), \ldots, tgt(\psi_i^m)), tgt(\psi)) < 2^{-n}$.
  Let $i := k + 1$. Then $\mathtt{dist}(f(tgt(\psi_i^1), \ldots, tgt(\psi_i^m)), f(t_1, \ldots, t_m)) =$ $\frac{1}{2} * max(\mathtt{dist}(tgt(\psi_i^1), t_1), \ldots, \mathtt{dist}(tgt(\psi_m^1), t_m)) < 2^{-n}$. Hence Lem. 5.1.25 yields $\mathtt{dist}(tgt(\psi), f(t_1, \ldots, t_m)) < 2^{-n}$. Thus we conclude.

Assume that the rule justifying $\psi \stackrel{\alpha}{\approx} \phi$ is $\mathsf{InfComp}$, so that $\psi = \cdot_{i<\omega} \psi_i$, $\phi = \cdot_{i<\omega} \phi_i$, and for all $i < \omega$, $\psi_i \stackrel{\alpha_i}{\approx} \phi_i$ where $\alpha_i < \alpha$.
Source terms: it is enough to apply IH on $\psi_0 \stackrel{\alpha_0}{\approx} \phi_0$ obtaining $src(\psi) = src(\psi_0) = src(\phi_0) = src(\phi)$.
Target terms and *mind*: Observe that IH can be applied on each $\psi_i \stackrel{\alpha_i}{\approx} \phi_i$, yielding $tgt(\psi_i) = tgt(\phi_i)$ and $mind(\psi_i) = mind(\phi_i)$. Then recalling the definitions of target and *mind* on ψ and ϕ suffices to conclude.

Assume that the rule used in the last step of the judgement $\psi \stackrel{\alpha}{\approx} \phi$ is Lim, so that for all $n < \omega$, $\psi \stackrel{\alpha_n}{\approx}_B \chi_n \cdot \psi'_n$ and $\phi \stackrel{\alpha_n}{\approx}_B \chi_n \cdot \phi'_n$, where $mind(\psi'_n) > n$, $mind(\phi'_n) > n$, $\alpha_n < \alpha$ and $\beta_n < \alpha$. Observe that $\stackrel{\alpha}{\approx}_B \subseteq \stackrel{\alpha}{\approx}$ for any ordinal α, so that IH can be applied to any premise of the Lim rule.
Source terms: applying IH on $\psi \stackrel{\alpha_0}{\approx} \chi_0 \cdot \psi'_0$ and $\phi \stackrel{\alpha_0}{\approx} \chi_0 \cdot \phi'_0$, we obtain $src(\psi) = src(\phi) = src(\chi_0)$.
Target terms: we prove $\mathtt{dist}(tgt(\psi), tgt(\phi)) = 0$. Let $n < \omega$. Then IH on $\psi \stackrel{\alpha_n}{\approx} \chi_n \cdot \psi'_n$ and $\phi \stackrel{\alpha_n}{\approx} \chi_n \cdot \phi'_n$ yields $tgt(\psi) = tgt(\psi'_n)$ and $tgt(\phi) = tgt(\phi'_n)$. Moreover, it is immediate to obtain $src(\psi'_n) = src(\phi'_n) = tgt(\chi_n)$. Recalling that $mind(\psi'_n) > n$ and $mind(\phi'_n) > n$, Lem. 5.2.17 can be applied to obtain $\mathtt{dist}(tgt(\chi_n), tgt(\psi)) = \mathtt{dist}(src(\psi'_n), tgt(\psi'_n)) < 2^{-n}$ and analogously $\mathtt{dist}(tgt(\chi_n), tgt(\phi)) = \mathtt{dist}(src(\phi'_n), tgt(\phi'_n)) < 2^{-n}$. Therefore Lem. 5.1.25 yields $\mathtt{dist}(tgt(\psi), tgt(\phi)) < 2^{-n}$. Thus we conclude.
Minimum activity depth: Assume for contradiction $n := mind(\psi) < mind(\phi)$. Observe $\psi \approx \chi_n \cdot \psi'_n$ and $\phi \approx \chi_n \cdot \phi'_n$, where $mind(\psi'_n) > n$ and $mind(\phi'_n) > n$. Then $mind(\psi) = n$ implies $mind(\chi_n) = n$, and therefore $mind(\phi) = n$, contradicting the assumption. The assertion $mind(\phi) < mind(\psi)$ can be contradicted analogously. Thus we conclude. □

The result about *mind* and *src* allows to prove that $\approx_B$ is closed w.r.t. the set of convergent proof terms.

Lemma 5.3.7. *Let ψ and ϕ proof terms such that $\psi \approx_B \phi$. Then ψ is a well-formed and convergent proof term iff ϕ is.*

Proof. We proceed by induction on α where $\psi \stackrel{\alpha}{\approx}_B \phi$, analysing the equational rule used in the last step in the corresponding $\approx_B$ derivation.

If the rule is Eqn, then we analyse the basic equation used.

- (IdLeft), i.e. $\psi = src(\phi) \cdot \phi$. It is immediate to verify the desired result.

- (IdRight), i.e. $\psi = \phi \cdot tgt(\phi)$. Observe that ϕ must be a convergent proof term. Thus we conclude immediately.

- (Assoc), i.e. $\psi = \chi \cdot (\xi \cdot \gamma)$ and $\phi = (\chi \cdot \xi) \cdot \gamma$. In this case, ψ is well-formed iff ϕ is well-formed iff χ, ξ and γ are well formed, and moreover χ and ξ are convergent. Moreover, ψ is convergent iff ϕ is convergent iff γ is convergent. Thus we conclude.

- (Struct), i.e. $\psi = f(\chi_1, \ldots, \chi_m) \cdot f(\xi_1, \ldots, \xi_m)$ and $\phi = f(\chi_1 \cdot \xi_1, \ldots, \chi_m \cdot \xi_m)$. In this case, ψ is well formed iff ϕ is well-formed iff all χ_i and ξ_i are well-formed, all the χ_i are also convergent (cfr. Lem. 5.2.20 for ψ), and $tgt(\chi_i) = src(\xi_i)$ for all i. Moreover, ψ is convergent iff all the ξ_i are convergent (cfr. again Lem. 5.2.20) iff all the $\chi_i \cdot \xi_i$ are convergent iff ϕ is convergent. Thus we conclude.

- (InfStruct), i.e. $\psi = \cdot_{i<\omega} f(\chi_i^1, \ldots, \chi_i^m)$ and $\phi = f(\cdot_{i<\omega} \chi_i^1, \ldots, \cdot_{i<\omega} \chi_i^m)$.

  $\Rightarrow$) Assume that ψ is well-formed and convergent. Given $n < \omega$, let $k_n < \omega$ be an index verifying $mind(f(\chi_i^1, \ldots \chi_i^m)) > n$ if $k_n < i$. Let j such that $1 \leqslant j \leqslant m$. Then for all $i < \omega$, $f(\chi_i^1, \ldots \chi_i^m)$ convergent implies χ_i^j convergent, cfr. Lem 5.2.20. In turn $src(f(\chi_{i+1}^1, \ldots \chi_{i+1}^m)) = tgt(f(\chi_i^1, \ldots \chi_i^m))$ implies immediately $src(\chi_{i+1}^j) = tgt(\chi_i^j)$. Finally, if $i > k_{n+1}$, then $mind(f(\chi_i^1, \ldots \chi_i^m)) > n+1$ implies $mind(\chi_i^j) > n$. Hence $\cdot_{i<\omega} \chi_i^j$ is well-formed and convergent. Consequently, so is ϕ.

  $\Leftarrow$) Assume that ϕ is well-formed and convergent. Given j such that $1 \leqslant j \leqslant m$ and $n < \omega$, let $k_{(n,j)}$ be an index verifying $mind(\psi_i^j) > n$ if $k_{(n,j)} < i$. Let $i < \omega$. Then χ_i^j convergent and $src(\psi_{i+1}^j) = tgt(\psi_i^j)$ for all j implies $f(\chi_i^1, \ldots, \chi_i^m)$ convergent and $src(f(\chi_{i+1}^1, \ldots \chi_{i+1}^m)) = tgt(f(\chi_i^1, \ldots \chi_i^m))$. Then ψ is a well-formed proof term. Moreover, for all $n < \omega$, if $i > max\{k_{(n,j)} \ / \ 1 \leqslant j \leqslant m\}$, then $mind(f(\chi_i^1, \ldots, \chi_i^m)) > n$. Consequently, ψ is convergent.

- (InOut), i.e. $\psi = \mu(\chi_1, \ldots, \chi_m)$ and $\phi = l[\chi_1, \ldots, \chi_m] \cdot \mu(t_1, \ldots, t_m)$. In this case, all χ_i are convergent proof terms, as it is explicitly noted in Dfn. 5.3.2. Then both ψ and ϕ are well-formed and convergent.

- (OutIn), i.e. $\psi = \mu(\chi_1, \ldots, \chi_m)$ and $\phi = \mu(s_1, \ldots, s_m) \cdot r[\chi_1, \ldots, \chi_m]$. In this case ψ is well-formed iff ϕ is well-formed iff χ_i are well-formed. Moreover, ψ is convergent iff ϕ is convergent iff all χ_i corresponding to variables occurring in the right-hand side r, which are exactly those occurring in $r[\chi_1, \ldots, \chi_m]$, are convergent; cfr. Lem. 5.2.22 and Lem. 5.2.21 respectively.

If the equational rule used in the last step of the derivation ending in $\psi \overset{\alpha}{\approx}_B \phi$ is Refl, Symm or Trans, then a straightforward argument suffices to conclude.

If the rule is Fun, Rule or Comp, then a simple argument based on Lem. 5.2.20, Lem 5.2.22 or just Dfn. 5.2.8 case (3) respectively, and IH, suffices to conclude.

Assume that the rule used in the last step of the derivation is InfComp. As the rule is symmetric, then it suffices to prove one side of the biconditional in the lemma statement. Then assume that $\psi = \cdot_{i<\omega} \psi_i$ is a well-formed and convergent proof term. Let $i < \omega$. Then ψ_i is convergent and $src(\psi_{i+1}) = tgt(\psi_i)$. Therefore IH implies convergence of ϕ_i, and Lem. 5.3.6 yields $src(\phi_{i+1}) = tgt(\phi_i)$. Hence ϕ is well-formed. Let $n < \omega$. Then convergence of ψ implies the existence of some $k_n < \omega$ verifying $mind(\psi_i) > n$ if $k_n < i$. In turn, Lem. 5.3.6 implies $mind(\phi_i) > n$ if $k_n < i$. Consequently, ψ is convergent. □

The following lemma shows that permutation equivalence is compatible with infinitary contexts.

Lemma 5.3.8. *Let C be a context having $k < \omega$ holes, and $\langle \psi_i \rangle_{i \leqslant k}$ and $\langle \phi \rangle_{i \leqslant k}$ two sequences of proof terms verifying $\psi_i \approx_B \phi_i$ for all i. Then $C[\psi_1, \ldots, \psi_k] \approx_B C[\phi_1, \ldots, \phi_k]$.*

Proof. An easy induction on $max\{|\mathsf{Bpos}(C, i)|\}$ suffices. Resort to the Fun equational rule for the inductive case. □

The following lemma shows that the (Struct) equation can be extended to contexts having a finite number of holes.

Lemma 5.3.9. *Let C be a context in Σ (i.e. built from function symbols only) having exactly $n < \omega$ occurrences of the box; and $\psi_1, \ldots, \psi_n$, $\phi_1, \ldots, \phi_n$ proof terms. Then $C[\psi_1, \ldots, \psi_n] \cdot C[\phi_1, \ldots, \phi_n] \approx_B C[\psi_1 \cdot \phi_1, \ldots, \psi_n \cdot \phi_n]$.*

Proof. We proceed by induction on $max(\{|\mathsf{Bpos}(C, i)|\})$.

If $C = \Box$, then we conclude immediately, notice that in this case $n = 1$.

Otherwise $C = f(C_1, \ldots, C_m)$. In this case

$C[\psi_1, \ldots, \psi_n] \cdot C[\phi_1, \ldots, \phi_n] =$
$\quad f(C_1[\psi_1, \ldots, \psi_{k1}], \ldots, C_m[\psi_{k(m-1)+1}, \ldots, \psi_n]) \cdot$
$\quad f(C_1[\phi_1, \ldots, \phi_{k1}], \ldots, C_m[\phi_{k(m-1)+1}, \ldots, \phi_n])$, and

$C[\psi_1 \cdot \phi_1, \ldots, \psi_n \cdot \phi_n] =$
$\quad f(C_1[\psi_1 \cdot \phi_1, \ldots, \psi_{k1} \cdot \phi_{k1}], \ldots, C_m[\psi_{k(m-1)+1} \cdot \phi_{k(m-1)+1}, \ldots, \psi_n \cdot \phi_n])$. We conclude by IH on each C_i, and then by the Fun equational rule. □

Lemma 5.3.10. *Let ψ be a trivial proof term. Then $\psi \approx src(\psi)$.*

Proof. Observe $\psi \approx_B src(\psi) \cdot \psi$ by (IdLeft). On the other hand, $src(\psi) \approx_B src(src(\psi)) \cdot src(\psi) = src(\psi) \cdot src(\psi)$, by (IdLeft) and Dfn. 5.2.4 respectively; recall that $src(\psi)$ is a trivial infinitary multistep. Moreover, for any $n < \omega$, $mind(\psi) = mind(src(\psi)) = \omega > n$, cfr. Lem. 5.2.11. Therefore the rule Lim can be applied to obtain $\psi \approx src(\psi)$. □

5.4 Denotation of reduction sequences

Proof terms are a means to describe different forms of contraction activity, comprising reduction sequences. A basic question, which is particularly relevant regarding the extension of the proof term model to infinitary rewriting we present in this chapter, arises: can *any* reduction sequence be faithfully described by means of a proof term?

To answer this question, we focus on proof terms which denote reduction sequences in a precise way, that is, reflecting the sequential nature of the activity denoted; we are not interested, in this section, in proof terms denoting simultaneous and/or localised contraction; cfr. Section 2.2.2. Formally, we define a proper subset of the set of valid proof terms, which we call *stepwise proof terms*, which include only (denotation of) single reduction steps and dots. A single reduction step is represented by a proof term including exactly one occurrence of a rule symbol and no occurrences of the dot; cfr. [BKdV03] Prop. 8.2.22:(i). We prove that any reduction sequence whose length is a countable ordinal can be denoted by means of a stepwise proof term. Observe that this result applies particularly to all convergent reduction sequences, cfr. Thm. 2 in [KdV05].

Once the possibility of denoting all countable-length reduction sequences is stated, the issue of *uniqueness of stepwise denotation* arises. As we discussed in the introduction to this chapter, cfr. Fig. 5.1 and the subsequent text, the fact that the dot is a binary symbol implies that different ways to associate a sequence of dots lead to different stepwise proof terms representing the same reduction sequence. Schematically, for a reduction sequence including three steps, say $\delta = a_1; a_2; a_3$, if ψ_i is a proof term representing a_i for $i = 1, 2, 3$, then $(\psi_1 \cdot \psi_2) \cdot \psi_3$ and $\psi_1 \cdot (\psi_2 \cdot \psi_3)$ are different stepwise representations of δ. Note that these proof terms are permutation equivalent. Moreover, they are related by the equivalence relation generated by the (Assoc) equation schema alone. This relation formalises the idea of "rebracketing", to wit, of changing the associativity of a sequence of dots. Let us give the name **rebracketing equivalence** to this relation. On the other hand, let us say that two stepwise proof terms are **denotationally equivalent** iff they denote the same reduction sequence.

These concepts allow to state the question about the uniqueness of denotation in a more precise way: **do denotational and rebracketing equivalences coincide**?

For the finitary case, a simple argument allows to verify that the answer to this question is positive. Note that the representation of a single step described above is unique. Therefore, by orienting the (Assoc) equation in either direction, *standard* denotations of reduction sequences can be obtained. These standard stepwise proof terms correspond with the result of coherently associating dots to the left or to the right.

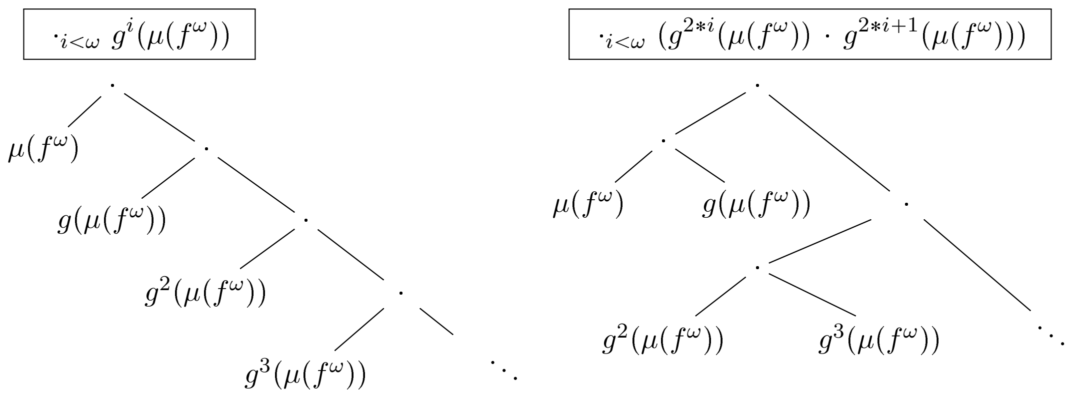


Figure 5.2: Two stepwise proof terms for the same infinite reduction sequence

For stepwise proof terms denoting infinite reduction sequences, the question seems less obvious. E.g. consider the rule $\mu : f(x) \to g(x)$, and the reduction sequence $\delta = f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega$. A simple way of organising the dots in a stepwise proof term denoting this sequence is considering the dot as right-associative; this criterion yields the proof term $\psi = \mu(f^\omega) \cdot (g(\mu(f^\omega)) \cdot (g^2(\mu(f^\omega)) \cdot (\ldots)))$, which can be noted concisely as $\cdot_{i<\omega}\ g^i(\mu(f^\omega))$. On the other hand, there may be other different ways to organise the same sequence of dots; one of them leads to the stepwise proof term $\phi = \cdot_{i<\omega}\ (g^{2*i}(\mu(f^\omega)) \cdot g^{2*i+1}(\mu(f^\omega)))$, where the steps are first grouped in pairs, and then right-associativity is applied to the set of *pairs* of steps. These proof terms are depicted in Fig. 5.2. We observe that for any $n < \omega$, it is easy to obtain, resorting to the (Assoc) equation schema only, that $\psi \approx_B (\mu(f^\omega) \cdot \ldots \cdot g^{2*n+1}(\mu(f^\omega))) \cdot g^{2*(n+1)}(\psi)$ and $\phi \approx_B (\mu(f^\omega) \cdot \ldots \cdot g^{2*n+1}(\mu(f^\omega))) \cdot g^{2*(n+1)}(\phi)$. Then we can obtain $\psi \approx \phi$ **by resorting to a limit argument**, i.e. by applying the Lim rule. We remark that we did not find a way to justify the permutation equivalence of these stepwise proof terms which do not involve the use of Lim.

In this section we prove that, provided the characterisation of permutation equivalence given in Sec. 5.3, denotational and rebracketing equivalences do coincide for infinitary term rewriting. The corresponding proofs make evident the role of the Lim rule in order to verify this assertion.

5.4.1 Stepwise proof terms

In the following, we introduce the set of stepwise proof terms, give some additional related definitions and state some basic properties of this subset of the set of valid proof terms.

Definition 5.4.1 (One-step)**.** *A* one-step *is an infinitary multistep including exactly one occurrence of a rule symbol. If ψ is a one-step, then we define the redex position of ψ, notation $\mathtt{rpos}(\psi)$, as the position of the unique rule symbol occurrence in ψ, and the depth of ψ, notation $d(\psi)$, as $|\mathtt{rpos}(\psi)|$; cfr. Dfn. 5.1.31 for the analogy with the corresponding notions as defined for a reduction step.*

Definition 5.4.2 (Stepwise proof term, Stepwise-or-nil proof term)**.** *A* stepwise proof term *is any proof term ψ whose formation satisfies any of the following conditions, where we refer to cases in Dfn. 5.2.8:*

- *ψ is a one-step, so it is built by case 1,*
- *ψ is built by case 2, so that $\psi = \cdot_{i<\omega}\ \psi_i$, and all of the ψ_i are stepwise proof terms, or*
- *ψ is built by case 3, so that $\psi = \psi_1 \cdot \psi_2$, and both ψ_1 and ψ_2 are stepwise proof terms.*

A stepwise-or-nil proof term *is any proof term ψ such that either ψ is a stepwise proof term or $\psi \in Ter^{\infty}(\Sigma)$.*

Definition 5.4.3 (Steps of a stepwise-or-nil proof term)**.** *For any ψ stepwise-or-nil proof term, we define the number of* steps *of ψ, notation $steps(\psi)$, as the countable ordinal defined as follows:*
if $\psi \in Ter^{\infty}(\Sigma)$, then $steps(\psi) := 0$.
if ψ is a one-step, then $steps(\psi) := 1$.
if $\psi = \cdot_{i<\omega}\ \psi_i$ then $steps(\psi) := \sum_{i<\omega} steps(\psi_i)$; cfr.Dfn. 5.1.1.
if $\psi = \psi_1 \cdot \psi_2$ then $steps(\psi) := steps(\psi_1) + steps(\psi_2)$.

Lemma 5.4.4. *Let ψ be a stepwise proof term, and let α the ordinal such that $\psi \in \mathbf{PT}_\alpha$. Then $steps(\psi)$ is a limit ordinal iff α is.*

Proof. Easy induction on α where $\psi \in \mathbf{PT}_\alpha$. □

Definition 5.4.5 (α-th component of a stepwise proof term)**.** *Let ψ be a stepwise proof term and α an ordinal such that $\alpha < steps(\psi)$. We define the α-*th component *of ψ, notation $\psi[\alpha]$, as the one-step defined as follows:*
if ψ is a one-step, then $\psi[0] := \psi$.
if $\psi = \cdot_{i<\omega}\ \psi_i$, then there are unique k and γ such that $\alpha = steps(\psi_0) + \ldots + steps(\psi_{k-1}) + \gamma$ and $\gamma < steps(\psi_k)$; cfr. Lem. 5.1.2. We define $\psi[\alpha] := \psi_k[\gamma]$.
if $\psi = \psi_1 \cdot \psi_2$ and $\alpha < steps(\psi_1)$ then $\psi[\alpha] := \psi_1[\alpha]$.
if $\psi = \psi_1 \cdot \psi_2$ and $steps(\psi_1) \leqslant \alpha$, then $\psi[\alpha] := \psi_2[\beta]$ such that $steps(\psi_1) + \beta = \alpha$.

Definition 5.4.6. *Let ψ be a stepwise proof term such that $steps(\psi) < \omega$. Then we define the* maximal depth activity *of ψ as $maxd(\psi) := max(d(\psi[n]) \;/\; n < steps(\psi))$. We also define the* maximal step depth *of ψ as $maxsd(\psi) := max(\mathtt{Pd}(\mu) \;/\; \mu \in R)$ where R is the set of all the rule symbols occurring in ψ.*

We show some expected properties of the components of a stepwise proof term. These properties particularly entail that a stepwise proof term can be seen as the concatenation of its components, so that the particular way in which they are associated is irrelevant.

Lemma 5.4.7. *Let ψ be a stepwise proof term, α an ordinal and $n < \omega$, such that $mind(\psi) > n$ and $\alpha < steps(\psi)$. Then*

1. $d(\psi[\alpha]) > n$.
2. $\mathtt{dist}(src(\psi[\alpha]), tgt(\psi[\alpha])) < 2^{-n}$.
3. $\mathtt{dist}(src(\psi), tgt(\psi[\alpha])) < 2^{-n}$.

Proof. We proceed by induction on ψ, cfr. Prop. 5.2.16. If ψ is a one-step then $\alpha = 0$ and $\psi[\alpha] = \psi$. Then we conclude immediately; cfr. Lemma 5.2.17 for (2) and (3).

Assume $\psi = \psi_1 \cdot \psi_2$. If $\alpha < steps(\psi_1)$, so that $\psi[\alpha] = \psi_1[\alpha]$, then we conclude by IH on ψ_1. Otherwise $\alpha = steps(\psi_1) + \beta$, so that $\psi[\alpha] = \psi_2[\beta]$. Then by applying IH on ψ_2 we obtain (1) and (2) immediately, and also $\mathtt{dist}(src(\psi_2), tgt(\psi[\alpha])) < 2^{-n}$. On the other hand we can apply Lemma 5.2.17 to ψ_1, obtaining $\mathtt{dist}(src(\psi), tgt(\psi_1)) < 2^{-n}$. Thus we conclude by Lemma 5.1.25 since $tgt(\psi_1) = src(\psi_2)$.

Assume $\psi = \cdot_{i<\omega}\, \psi_i$. Let k, β such that $\psi[\alpha] = \psi_k[\beta]$, so that $\beta < steps(\psi_k)$. Then IH on ψ_k yields immediately (1) and (2), and also $\mathtt{dist}(src(\psi_k), tgt(\psi[\alpha])) < 2^{-n}$. On the other hand, for each $i < k$ it is immediate that $mind(\psi_i) \geqslant mind(\psi) > n$, then an easy induction on k using Lemma 5.2.17 and Lemma 5.1.25 yields $\mathtt{dist}(src(\psi), src(\psi_k)) < 2^{-n}$. Thus we conclude by Lemma 5.1.25. □

Lemma 5.4.8. *Let ψ be a convergent stepwise proof term such that $mind(\psi) > p$, and $\alpha < steps(\psi)$. Then $\mathtt{dist}(tgt(\psi[\alpha]), tgt(\psi)) < 2^{-p}$.*

Proof. We proceed by induction on ψ. If ψ is a one-step then $\alpha = 0$ and it suffices to observe that $\psi[0] = \psi$.

Assume $\psi = \psi_1 \cdot \psi_2$. If $\alpha < steps(\psi_1)$, then IH on ψ_1 yields $\mathtt{dist}(tgt(\psi[\alpha]), tgt(\psi_1)) < 2^{-p}$. On the other hand, Lemma 5.2.17 implies $\mathtt{dist}(src(\psi_2), tgt(\psi)) < 2^{-p}$. We conclude by Lemma 5.1.25 since $tgt(\psi_1) = src(\psi_2)$. Otherwise, $\alpha = steps(\psi_1) + \beta$, then $\psi[\alpha] = \psi_2[\beta]$. In this case we can apply IH on ψ_2 obtaining $\mathtt{dist}(tgt(\psi_2[\beta]), tgt(\psi_2)) < 2^{-p}$, thus we conclude.

Assume $\psi = \cdot_{i<\omega}\, \psi_i$ and let k, γ such that $\psi[\alpha] = \psi_k[\gamma]$. Then IH on ψ_k yields $\mathtt{dist}(tgt(\psi[\alpha]), tgt(\psi_k)) < 2^{-p}$. Moreover, Lemma 5.2.17 on $\cdot_{i<\omega}\, \psi_{k+1+i}$ implies $\mathtt{dist}(src(\psi_{k+1}), tgt(\psi)) < 2^{-p}$. Thus we conclude by Lemma 5.1.25. □

Lemma 5.4.9. *Let ψ be a stepwise proof term. Then $src(\psi[0]) = src(\psi)$.*

Proof. Easy induction on ψ. □

Lemma 5.4.10. *Let ψ be a stepwise proof term such that $steps(\psi) = \alpha + 1$. Then $tgt(\psi) = tgt(\psi[\alpha])$.*

Proof. We proceed by induction on ψ. If ψ is a one-step then $\alpha = 0$ and we conclude immediately.

Assume $\psi = \psi_1 \cdot \psi_2$. Then $\alpha < steps(\psi_1)$ would imply $\alpha+1 = steps(\psi) \leqslant steps(\psi_1)$, which is not possible since $steps(\psi_2) > 0$. Then let β be the ordinal verifying $steps(\psi_1) + \beta = \alpha$, so that $\psi[\alpha] = \psi_2[\beta]$. We observe that $steps(\psi_1) + \beta + 1 = \alpha + 1 = steps(\psi)$, then $steps(\psi_2) = \beta + 1$. We conclude by IH on ψ_2.

Finally, $\psi = \cdot_{i<\omega}\ \psi_i$ contradicts $steps(\psi)$ to be a successor ordinal. Thus we conclude. □

Lemma 5.4.11. *Let ψ be a convergent stepwise proof term such that $steps(\psi)$ is a limit ordinal. Then $tgt(\psi) = \lim_{\alpha \to steps(\psi)} tgt(\psi[\alpha])$.*

Proof. Observe $steps(\psi)$ being a limit ordinal implies $\psi = \cdot_{i<\omega}\ \psi_i$ (cfr. Lem. 5.4.4 and Lem. 5.2.12), so that $tgt(\psi)$ is defined to be equal to $\lim_{i\to\omega} tgt(\psi_i)$. Observe that Lem 5.2.17:(a) implies this limit to be defined. Let $p \in \mathbb{N}$, let k' such that $k' < j < \omega$ implies $\mathtt{dist}(tgt(\psi_j), tgt(\psi)) < 2^{-p}$, k'' such that $mind(\psi_j) > p$ if $j > k''$, and $k := max(k', k'')$.

Let $\beta = steps(\psi_0) + \ldots + steps(\psi_k)$ and $\gamma > \beta$. Then $\gamma = steps(\psi_0) + \ldots + steps(\psi_j) + \gamma'$ where $\gamma' < steps(\psi_{j+1})$ and $j \geqslant k$, so that $\psi[\gamma] = \psi_{j+1}[\gamma']$. Then $j + 1 > k \geqslant k''$, so that Lemma 5.4.8 implies $\mathtt{dist}(tgt(\psi[\gamma]), tgt(\psi_{j+1})) < 2^{-p}$. On the other hand, $j + 1 > k \geqslant k'$ implies $\mathtt{dist}(tgt(\psi_{j+1}), tgt(\psi)) < 2^{-p}$. Hence Lemma 5.1.25 yields $\mathtt{dist}(tgt(\psi[\gamma]), tgt(\psi)) < 2^{-p}$. Consequently, we conclude. □

Lemma 5.4.12. *Let ψ be a stepwise proof term and $\alpha < steps(\psi)$ such that $\alpha = \alpha' + 1$. Then $src(\psi[\alpha]) = tgt(\psi[\alpha'])$.*

Proof. We proceed by induction on ψ. Observe ψ is a one-step would imply $\alpha = 0$, contradicting $\alpha = \alpha' + 1$.

Assume $\psi = \psi_1 \cdot \psi_2$. We consider three cases

- If $\alpha < steps(\psi_1)$ then we conclude just by IH on ψ_1.
- If $\alpha = steps(\psi_1)$, then $\psi[\alpha] = \psi_2[0]$ and $\psi[\alpha'] = \psi_1[\alpha']$ where $\alpha' + 1 = \alpha = steps(\psi_1)$. Then $tgt(\psi[\alpha']) = tgt(\psi_1)$ and $src(\psi[\alpha]) = src(\psi_2)$, by Lemma 5.4.10 and Lemma 5.4.9 respectively. Thus we conclude.
- If $\alpha > steps(\psi_1)$, then $\alpha' = steps(\psi_1) + \beta'$ and $\alpha = steps(\psi_1) + (\beta' + 1)$, therefore $\psi[\alpha] = \psi_2[\beta' + 1]$ and $\psi[\alpha'] = \psi_2[\beta']$. Observe that $\alpha < steps(\psi)$ implies $\beta' + 1 < steps(\psi_2)$. Hence we conclude by IH on ψ_2.

Assume $\psi = \cdot_{i<\omega}\ \psi_i$. Let k, γ such that $\alpha = steps(\psi_0) + \ldots + steps(\psi_{k-1}) + \gamma$ and $\gamma < steps(\psi_k)$, so that $\psi[\alpha] = \psi_k[\gamma]$. If $\gamma = 0$, then $steps(\psi_{k-1}) = \beta + 1$ for some β, and $\alpha' = steps(\psi_0) + \ldots + steps(\psi_{k-2}) + \beta$, so that $\psi[\alpha'] = \psi_{k-1}[\beta]$. Therefore $src(\psi[\alpha]) = src(\psi_k)$ and $tgt(\psi[\alpha']) = tgt(\psi_{k-1})$, by Lemma 5.4.9 and Lemma 5.4.10 respectively. Thus we conclude. Otherwise $\gamma = \gamma' + 1$; notice that γ being a limit ordinal would contradict α being a successor one. In this case $\psi[\alpha'] = \psi_k[\gamma']$, thus we conclude by IH on ψ_k. □

Lemma 5.4.13. *Let ψ be a stepwise proof term. Then*

$$\begin{array}{rcl} mind(\psi) & = & min(d(\psi[\alpha]) \ / \ \alpha < steps(\psi)) \\ & = & min(mind(\psi[\alpha]) \ / \ \alpha < steps(\psi)) \end{array}$$

Proof. We prove that $mind(\psi) = min(mind(\psi[\alpha]) \ / \ \alpha < steps(\psi))$. The rest of the statement follows immediately since it is trivial to verify $d(\psi[\alpha]) = mind(\psi[\alpha])$ for any α; cfr. Dfn. 5.2.6.

We proceed by induction on ψ; cfr. Prop. 5.2.16. We define $mind'(\psi) := min(mind(\psi[\alpha]) \ / \ \alpha < steps(\psi))$, so we must verify $mind(\psi) = mind'(\psi)$. If ψ is a one-step then the result holds immediately.

Assume $\psi = \psi_1 \cdot \psi_2$. In this case, IH on ψ_i yields $mind(\psi_i) = mind'(\psi_i)$ for each $i = 1, 2$, and Dfn. 5.2.8 implies $mind(\psi) = min(mind(\psi_1), mind(\psi_2))$. Then it suffices to verify $mind'(\psi) = min(mind'(\psi_1), mind'(\psi_2))$. From the definition of $mind'$, it is immediate that $mind'(\psi) \leqslant mind'(\psi_i)$ for $i = 1, 2$. Assume $mind'(\psi_1) \leqslant mind'(\psi_2)$. Notice $mind'(\psi) < mind'(\psi_1)$ would imply the existence of some γ verifying $mind'(\psi[\gamma]) < mind'(\psi_1)$, contradicting either the definition of $mind'(\psi_1)$ (if $\gamma < steps(\psi_1)$) or the assertion $mind'(\psi_1) \leqslant mind'(\psi_2)$ (otherwise). Hence $mind'(\psi) = mind'(\psi_1)$. A similar argument for the case $mind'(\psi_2) < mind'(\psi_1)$ is enough to conclude.

If $\psi = \cdot_{i<\omega} \psi_i$, then an argument similar to that used for binary composition applies. To verify that $mind'(\psi) = min_{i<\omega}(mind'(\psi_i))$, observe that $mind'(\psi) \leqslant mind'(\psi_i)$ for all i, and consider n such that $mind'(\psi_n) \leqslant mind'(\psi_i)$ for all i. Then we can contradict $mind'(\psi) < mind'(\psi_n)$ proceeding as in the previous case, hence $mind'(\psi) = mind'(\psi_n)$. Thus we conclude. □

5.4.2 Denotation – formal definition and proof of existence

In this section, we formalise the notion of a stepwise-or-nil proof term *denoting* a reduction sequence, resorting to the definitions of length and α-th component of stepwise-or-nil proof terms, given in the presentation of such terms. Then we prove the existence, for any reduction sequence having a countable ordinal length, of a stepwise-or-nil proof term which denotes it.

As we have discussed in the introduction to Section 5.4, denotation of a reduction sequence is not unique. In the next section, we will investigate how to characterise the proof terms denoting the same reduction sequence.

Definition 5.4.14 (Denotation for reduction steps)**.** *Let $a = \langle t, p, \mu \rangle$ be a reduction step, and ψ a one-step. Then ψ* denotes *a iff all the following apply: $src(\psi) = t$, $tgt(\psi) = tgt(a)$, and $\psi(p) = \mu$, therefore $d(a) = mind(\psi)$.*

Definition 5.4.15 (Mapping from one-steps to reduction steps)**.** *Let T be a TRS. We define the mapping* sden *from the set of one-steps for T to the set of reduction steps for T, as follows:* $\mathsf{sden}(\psi) := \langle src(\psi), \mathtt{rpos}(\psi), \psi(\mathtt{rpos}(\psi)) \rangle$.

Lemma 5.4.16. *Let ψ be a one-step and a a reduction step. Then ψ denotes a iff $a = \mathsf{sden}(\psi)$.*

Proof. We prove each direction of the biconditional.
$\Rightarrow$): Let us say $a = \langle t, p, \mu \rangle$. Hypotheses imply immediately $t = src(\psi)$, and also $\psi(p) = \mu$, so that $p = \mathtt{rpos}(\psi)$ and $\mu = \psi(\mathtt{rpos}(\psi))$. Thus we conclude. $\Leftarrow$): Let us say $\mathsf{sden}(\psi) = \langle t, p, \mu \rangle$ and $\mu : l \to h$. Then it is immediate from Dfn. 5.4.15 to verify $src(\psi) = t$ and $\psi(p) = \mu$. In turn, observe that $tgt(\psi) = \psi[h[t_1, \ldots, t_m]]_p$ where $\psi|_p = \mu(t_1, \ldots, t_m)$, and $t = src(\psi) = \psi[l[t_1, \ldots, t_m]]_p$, so that it is straightforward to verify $tgt(\mathsf{sden}(\psi)) = tgt(\psi)$. Thus we conclude. □

Definition 5.4.17 (Denotation for reduction sequences)**.** *Let δ be a reduction sequence, and ψ a stepwise-or-nil proof term. We will say that ψ* denotes *δ iff $steps(\psi) = \mathsf{length}(\delta)$, $src(\psi) = src(\delta)$ and $\psi[\alpha]$ denotes $\delta[\alpha]$ for all $\alpha < \mathsf{length}(\delta)$.*

Lemma 5.4.18. *Let δ be a reduction sequence, and ψ a stepwise-or-nil proof term, such that ψ denotes δ. Then $mind(\psi) = mind(\delta)$, ψ is convergent iff δ is, and in that case, $tgt(\psi) = tgt(\delta)$.*

Proof. If $\psi \in Ter^{\infty}(\Sigma)$, then the result holds immediately.

Otherwise, the result about *mind* stems immediately from Lem. 5.4.13.

We prove the result about convergence. Assume that $steps(\psi)$ is a limit ordinal, then $\psi = \cdot_{i<\omega} \psi_i$; cfr. Lem. 5.4.4 and Lem. 5.2.12. Assume δ convergent, consider some $k < \omega$, and α such that $d(\delta[\beta]) > k$ if $\beta > \alpha$. Lem. 5.1.2 implies that $\alpha = \sum_{i<n} steps(\psi_i) + \gamma$ and $\gamma < steps(\psi_n)$ for some n; so that $\alpha < \sum_{i\leqslant n} steps(\psi_i)$. Consider $j > n$, and $\gamma < steps(\psi_j)$. Observe $\psi_j[\gamma] = \psi[\beta]$ where $\beta = \sum_{i<j} steps(\psi_i) + \gamma$, so that $\beta \geqslant \sum_{i\leqslant n} steps(\psi_i) > \alpha$. Therefore $mind(\psi_j[\gamma]) = mind(\psi[\beta]) = d(\delta[\beta]) > k$. Hence Lem. 5.4.13 implies that $mind(\psi_j) > k$. Consequently, ψ is convergent.

Conversely, assume ψ convergent, let $k < \omega$, consider $n < \omega$ such that $mind(\psi_j) > k$ if $j > n$. Let $\alpha := \sum_{i\leqslant n} steps(\psi_i)$, and take β such that $\alpha < \beta < \mathsf{length}(\delta)$. Then Lem. 5.1.2 implies $\beta = \sum_{i<j} steps(\psi_i) + \gamma$ and $\gamma < steps(\psi_j)$, moreover, $\beta > \alpha$ implies $j > n$. Hence $d(\delta[\beta]) = mind(\psi_j[\gamma]) > k$ by Lem. 5.4.13. Consequently, the requirement about depths in the characterisation of convergent reduction sequences, i.e. condition (2c) in Dfn. 5.1.32, holds for δ. To prove the existence of $\lim_{\alpha\to\mathsf{length}(\delta)} tgt(\delta[\alpha])$, i.e. condition (2a) in Dfn. 5.1.32, it suffices to observe that Lem. 5.2.17:(a) implies that $tgt(\psi)$ is defined, and in turn Lem. 5.4.11 implies the desired limit to equal $tgt(\psi)$. Hence δ is convergent.

If $steps(\psi)$ is a successor ordinal, then assuming δ is convergent, a straightforward induction on ψ suffices to prove that ψ is convergent as well; observe that Lem. 5.4.4 and Lem 5.2.12 imply that only one-step and binary concatenation must be considered. For the other direction, it is enough to observe that $\mathsf{length}(\delta)$ being a successor ordinal implies immediately convergence of δ.

Finally, the result about targets stems immediately from Lem. 5.4.11 and Lem. 5.4.10. □

Proposition 5.4.19. *Let δ be a reduction sequence having a countable length. Then there exists a stepwise-or-nil proof term ψ such that ψ denotes δ.*

Proof. We proceed by induction on $\mathsf{length}(\delta)$.

If $\mathsf{length}(\delta) = 0$, i.e. $\delta = \mathsf{Id}_t$, then it suffices to take $\psi := t$.

Assume that $\mathsf{length}(\delta) = 1$. Let us say $\delta[0] = \langle t, p, \mu \rangle$ where $\mu : l \to h$, implying that $t|_p = l[t_1, \ldots, t_m]$. Take $\psi := t[\mu(t_1, \ldots, t_m)]_p$. It is immediate to verify that ψ is a stepwise proof term verifying $steps(\psi) = 1$. Moreover, a simple analysis yields $src(\psi) = src(\delta[0]) = src(\delta) = t$. Furthermore, $\psi(p) = \mu$, and $tgt(\psi) = tgt(\delta[0]) = t[h[t_1, \ldots, t_m]]_p$; therefore $\psi[0] = \psi$ denotes $\delta[0]$. Hence ψ denotes δ.

Assume $\mathsf{length}(\delta) = \alpha + 1$ and $\alpha > 0$. In this case, applying twice IH yields the existence of ψ_1, ψ_2 such that ψ_1 denotes $\delta[0, \alpha)$ and ψ_2 denotes $\delta[\alpha, \alpha + 1)$. Then a straightforward analysis allows to obtain that $\psi := \psi_1 \cdot \psi_2$ denotes δ.

Assume $\alpha := \mathsf{length}(\delta)$ is a limit ordinal; recall that α is countable. Then Prop. 5.1.3 implies $\alpha = \sum_{i<\omega} \alpha_i$ where $\alpha_i < \alpha$ for all $i < \omega$. Therefore, for any $n < \omega$, IH can be applied to obtain some ψ_n denoting $\delta[\sum_{i<n} \alpha_i, \sum_{i\leqslant n} \alpha_i)$. We take $\psi := \cdot_{i<\omega} \psi_i$.

Let $n < \omega$. It is easy to verify that $\delta[\sum_{i<n} \alpha_i, \sum_{i\leqslant n} \alpha_i)$ is convergent, then Lem. 5.4.18 implies $tgt(\psi_n) = tgt(\delta[\sum_{i<n} \alpha_i, \sum_{i\leqslant n} \alpha_i)) = src(\delta[\sum_{i\leqslant n} \alpha_i, \sum_{i\leqslant n+1} \alpha_i) = src(\psi_{n+1})$; cfr. conditions about sources and targets in Dfn. 5.1.32. Hence ψ is a well-formed proof term. Recalling that $\mathsf{length}(\delta[\sum_{i<n} \alpha_i, \sum_{i\leqslant n} \alpha_i)) = \alpha_n$, it is straightforward to obtain $steps(\psi) = \mathsf{length}(\delta) = \alpha$. Moreover, $src(\psi) = src(\psi_0) = src(\delta[0, \alpha_0)) = src(\delta)$, recall that ψ_0 denotes $\delta[0, \alpha_0)$. Let $\beta < \alpha$. Then Lem. 5.1.2 implies the existence of unique k and γ such that $\beta = \sum_{i<k} \alpha_i + \gamma$ and $\gamma < \alpha_k$. Therefore $\psi[\beta] = \psi_k[\gamma]$ and $\delta[\beta] = \delta[\sum_{i<k} \alpha_i, \sum_{i\leqslant k} \alpha_i)[\gamma]$, cfr. Dfn. 5.4.5 and Dfn. 5.1.39. Hence ψ_k denoting $\delta[\sum_{i<k} \alpha_i, \sum_{i\leqslant k} \alpha_i)$ implies that $\psi[\beta]$ denotes $\delta[\beta]$. Consequently, we conclude. □

We remark that Prop. 5.4.19 gives a positive answer to the question put at the beginning of Section 5.4: *any* reduction sequence whose length is a countable ordinal, thus including particularly all the strongly convergent reduction sequences, can be faithfully denoted in the model of infinitary rewriting based on proof terms we propose in this chapter.

5.4.3 Uniqueness of denotation

In this section we will prove the claim we made at the beginning of Section 5.4: *rebracketing equivalence*, which is the result of restricting the permutation equivalence relation introduced in Section 5.3 by allowing only the instances of the (Assoc) schema as basic equations, is an adequate syntactic counterpart of the relation of "denoting the same reduction sequence", i.e. *denotational equivalence*, between stepwise proof terms.

In the following we will give formal definitions for the concepts of denotational and rebracketing equivalence, and subsequently prove that the defined relations coincide.

Definition 5.4.20 (Denotational equivalence)**.** *Let ψ, ϕ be stepwise-or-nil proof terms. We say that ψ and ϕ are* denotationally equivalent*, notation $\psi \equiv \phi$, iff either $steps(\psi) = steps(\phi) = 0$ and $\psi = \phi$, or $steps(\psi) = steps(\phi) > 0$ and $\psi[\alpha] = \phi[\alpha]$ for all $\alpha < steps(\psi)$.*

Definition 5.4.21 (Layer of rebracketing equivalence)**.** *Let α be a countable ordinal.*

We define the α-th level of base rebracketing equivalence *relation, notation $\overset{\alpha}{\approx}_{(B)}$, on the set of stepwise-or-nil proof terms, as follows. Given ψ and ϕ stepwise-or-nil proof terms, $\psi \overset{\alpha}{\approx}_{(B)} \phi$ iff the equation $\psi \overset{\alpha}{\sim} \phi$ can be obtained by means of the equational logic system whose basic equations are the valid instances of the* (Assoc) *equation schema described in Dfn. 5.3.2, and whose equational rules are* Refl*,* Eqn*,* Symm*,* Trans*,* Comp *and* InfComp*, described also in Dfn. 5.3.2.*

We also define the α-th level of rebracketing equivalence *relation, notation $\overset{\alpha}{\approx}_{()}$, on the set of stepwise-or-nil proof terms, analogously, the only difference being that a rule is added, namely the version of the* Lim *rule which results from changing, in the premises, the references to the $\overset{\alpha_k}{\approx}_B$ and $\overset{\beta_k}{\approx}_B$ relations, to $\overset{\alpha_k}{\approx}_{(B)}$ and $\overset{\beta_n}{\approx}_{(B)}$ respectively.*

Definition 5.4.22 (Rebracketing equivalence)**.** *Let ψ, ϕ be stepwise-or-nil proof terms. We say that ψ and ϕ are (base) rebracketing equivalent, notation ($\psi \approx_{(B)} \phi$) $\psi \approx_{()} \phi$, iff ($\psi \overset{\alpha}{\approx}_{(B)} \phi$) $\psi \overset{\alpha}{\approx}_{()} \phi$ for some $\alpha < \omega_1$.*

Observe that all the following inclusions hold where α is any countable ordinal: $\overset{\alpha}{\approx}_{(B)} \subseteq \overset{\alpha}{\approx}_{()}$, $\overset{\alpha}{\approx}_{(B)} \subseteq \overset{\alpha}{\approx}_B$, $\overset{\alpha}{\approx}_{()} \subseteq \overset{\alpha}{\approx}$, and consequently $\approx_{(B)} \subseteq \approx_{()}$, $\approx_{(B)} \subseteq \approx_B$ and $\approx_{()} \subseteq \approx$.

Therefore, several results stated for permutation equivalence hold also for rebracketing equivalence. Particularly, properties proved for the $\approx_B$ relation also apply to $\approx_{(B)}$.

Lemma 5.4.23. *Let ψ a stepwise proof term, and α such that $\psi \in \mathbf{PT}_\alpha$. Then there exists $n < \omega$ such that $\alpha = steps(\psi) + n$. Moreover, if α is a limit ordinal, then $n = 0$, i.e. $\alpha = steps(\psi)$.*

Proof. We proceed by induction on α. If $\alpha = 1$ then ψ is a one-step, and then $steps(\psi) = 1 = \alpha$.

Assume α is a successor ordinal and $\alpha > 1$. In this case, Lem. 5.2.12 and Lem. 5.2.13 imply that $\psi = \psi_1 \cdot \psi_2$, $\psi_i \in \mathbf{PT}_{\alpha_i}$ for $i = 1.2$, α_2 is successor, and $\alpha = \alpha_1 + \alpha_2 + 1$. IH implies $\alpha_1 = steps(\psi_1) + n_1$ and $\alpha_2 = steps(\psi_2) + n_2$. If $steps(\psi_2) < \omega$, then $\alpha = steps(\psi) + n_1 + n_2 + 1$, otherwise $\alpha = steps(\psi) + n_2 + 1$. In either case the conclusion holds, thus we conclude.

Assume that α is a limit ordinal, so that Lem. 5.2.12 implies $\psi = \cdot_{i<\omega} \psi_i$ and $\alpha = \sum_{i<\omega} \alpha_i$ where $\psi_i \in \mathbf{PT}_{\alpha_i}$ for all $i < \omega$. Observe $\alpha_i < \alpha$ for all i. Then we can apply IH on each i obtaining $\alpha_i = steps(\psi_i) + n_i$, so that proving $\sum_{i<\omega} steps(\psi_i) + n_i = \sum_{i<\omega} steps(\psi_i)$ suffices to conclude.

Let $k < \omega$. Observe $\sum_{i<k} steps(\psi_i) + n_i \leqslant \sum_{i<k} steps(\psi_i) + \sum_{i<k} n_i < \sum_{i<k} steps(\psi_i) + \omega$. On the other hand, $\sum_{i<\omega} steps(\psi_i) = \sum_{i<k} steps(\psi_i) + \sum_{i<\omega} steps(\psi_{k+i}) \geqslant \sum_{i<k} steps(\psi_i) + \omega$. Then $\sum_{i<k} steps(\psi_i) + n_i < \sum_{i<\omega} steps(\psi_i)$. Consequently, $\sum_{i<\omega} steps(\psi_i) + n_i \leqslant \sum_{i<\omega} steps(\psi_i)$. We conclude by observing that it is straightforward to obtain $\sum_{i<\omega} steps(\psi_i) \leqslant \sum_{i<\omega} steps(\psi_i) + n_i$. □

Lemma 5.4.24. *Let ψ be a stepwise proof term. Then $steps(\psi)$ is a limit ordinal iff ψ is an infinite concatenation.*

Proof. We proceed by induction on α where $\psi \in \mathbf{PT}_\alpha$; cfr. Dfn. 5.3.2. If ψ is a one-step, then we conclude immediately. If $\psi = \psi_1 \cdot \psi_2$ and it is not an infinite concatenation, then ψ_2 is neither. Therefore we can apply IH on ψ_2 obtaining that $steps(\psi_2)$ is a successor ordinal. We conclude by recalling that $steps(\psi) = steps(\psi_1) + steps(\psi_2)$. Finally, if ψ is an infinite concatenation, then Lem. 5.2.12 implies that $\psi \in \mathbf{PT}_\alpha$ where α is a limit ordinal. In turn, Lem. 5.4.23 implies that $steps(\psi) = \alpha$. □

Lemma 5.4.25. *Let ψ be a stepwise proof term, α an ordinal verifying $0 < \alpha < steps(\psi)$, and β such that $\psi \in \mathbf{PT}_\beta$. Then there exist ϕ, χ such that $\psi \approx_{(B)} \phi \cdot \chi$ and $steps(\phi) = \alpha$. Moreover, if $\phi \in \mathbf{PT}_\gamma$ and $\chi \in \mathbf{PT}_\delta$, then $\gamma < \beta$ and $\delta \leqslant \beta$.*

Proof. We proceed by induction on ψ.

If $\psi \in Ter^\infty(\Sigma)$ or ψ is a one-step, then no α verifies the hypotheses.

Assume $\psi = \psi_1 \cdot \psi_2$, so that $\beta = \beta_1 + \beta_2 + 1$ where $\psi_i \in \mathbf{PT}_{\beta_i}$ for $i = 1, 2$.

- If $steps(\psi_1) < \alpha$, so that $\alpha = steps(\psi_1) + \alpha'$, then IH on ψ_2 yields the existence of ϕ_2, χ_2 satisfying $\psi_2 \approx_{(B)} \phi_2 \cdot \chi_2$, $steps(\phi_2) = \alpha'$, $\gamma_2 < \beta_2$ and $\delta \leqslant \beta_2$, where $\phi_2 \in \mathbf{PT}_{\gamma_2}$ and $\chi_2 \in \mathbf{PT}_\delta$.

  Therefore, $\psi \approx_{(B)} \psi_1 \cdot (\phi_2 \cdot \chi_2) \approx_{(B)} (\psi_1 \cdot \phi_2) \cdot \chi_2$ and $steps(\psi_1 \cdot \phi_2) = steps(\psi_1) + \alpha' = \alpha$. Moreover, $\psi_1 \cdot \phi_2 \in \mathbf{PT}_\gamma$ where $\gamma = \beta_1 + \gamma_2 + 1 < \beta_1 + \beta_2 + 1 = \beta$, and $\delta \leqslant \beta_2 < \beta$.

- If $steps(\psi_1) = \alpha$ then the result holds trivially.

- If $steps(\psi_1) > \alpha$, then IH on ψ_1 yields $\psi_1 \approx_{(B)} \phi_1 \cdot \chi_1$, $steps(\phi_1) = \alpha$, $\gamma < \beta_1$ and $\delta_1 \leqslant \beta_1$, where $\phi_1 \in \mathbf{PT}_\gamma$ and $\chi_1 \in \mathbf{PT}_{\delta_1}$.

  Therefore $\psi \approx_{(B)} (\phi_1 \cdot \chi_1) \cdot \psi_2 \approx_{(B)} \phi_1 \cdot (\chi_1 \cdot \psi_2)$. Moreover, $\gamma < \beta_1 < \beta$, and $\chi_1 \cdot \psi_2 \in \mathbf{PT}_\delta$ where $\delta = \delta_1 + \beta_2 + 1 \leqslant \beta_1 + \beta_2 + 1 = \beta$.

Assume $\psi = \cdot_{i<\omega} \psi_i$, so that $steps(\psi) = \sum_{i<\omega} steps(\psi_i)$. In this case, Lem 5.2.12 and Lem 5.4.23 imply that β is a limit ordinal, and therefore $\beta = steps(\psi)$. Moreover, Lem 5.1.2 implies $\alpha = \sum_{i<n} steps(\psi_i) + \alpha'$ where $\alpha' < steps(\psi_n)$, for some n and α'. IH on ψ_n yields $\psi_n \approx_{(B)} \phi_n \cdot \chi_n$ such that $steps(\phi_n) = \alpha'$; observe that $steps(\chi_n) \leqslant steps(\psi_n)$. Therefore

$$\begin{array}{lll}\psi & \approx_{(B)} & ((\psi_0 \cdot \ldots \cdot \psi_{n-1}) \cdot \psi_n) \cdot \cdot_{i<\omega} \psi_{n+1+i} \\ & \approx_{(B)} & ((\psi_0 \cdot \ldots \cdot \psi_{n-1}) \cdot (\phi_n \cdot \chi_n)) \cdot \cdot_{i<\omega} \psi_{n+1+i} \\ & \approx_{(B)} & ((\psi_0 \cdot \ldots \cdot \psi_{n-1} \cdot \phi_n) \cdot \chi_n) \cdot \cdot_{i<\omega} \psi_{n+1+i} \\ & \approx_{(B)} & (\psi_0 \cdot \ldots \cdot \psi_{n-1} \cdot \phi_n) \cdot (\chi_n \cdot \cdot_{i<\omega} \psi_{n+1+i})\end{array}$$

where $steps(\psi_0 \cdot \ldots \cdot \psi_{n-1} \cdot \phi_n) = \sum_{i<n} steps(\psi_i) + \alpha' = \alpha$.

Moreover, if $\psi_0 \cdot \ldots \cdot \psi_{n-1} \cdot \phi_n \in \mathbf{PT}_\gamma$, then Lem. 5.4.23 implies the existence of some $k < \omega$ such that $\gamma = steps(\psi_0) + \ldots + steps(\psi_{n-1}) + \alpha' + k < steps(\psi_0) + \ldots + steps(\psi_{n-1}) + steps(\psi_n) + \omega \leqslant steps(\psi) = \beta$. On the other hand, notice that $\chi_n \cdot \cdot_{i<\omega} \psi_{n+1+i}$ is an infinitary concatenation, so that $\chi_n \cdot \cdot_{i<\omega} \psi_{n+1+i} \in \mathbf{PT}_\delta$ implies δ to be a limit ordinal; cfr. Lem. 5.2.12. Therefore, recalling that $steps(\chi_n) \leqslant steps(\psi_n)$, Lem. 5.4.23 yields $\delta = steps(\chi_n) + \sum_{i<\omega} steps(\psi_{n+1+i}) \leqslant \sum_{i<\omega} steps(\psi_{n+i}) \leqslant steps(\psi) = \beta$. □

Lemma 5.4.26. *Let $\psi \equiv \phi$, such that both are convergent. Then $tgt(\psi) = tgt(\phi)$.*

Proof. Easy, cfr. Lem. 5.4.10 and Lem 5.4.11. □

Lemma 5.4.27. *Let $\psi \cdot \phi \equiv \psi' \cdot \phi'$ and $\psi \equiv \psi'$. Then $\phi \equiv \phi'$.*

Proof. Observe that definition of stepwise proof terms implies that $steps(\phi) > 0$ and $steps(\phi') > 0$. Given $steps(\psi \cdot \phi) = steps(\psi' \cdot \phi')$ and $steps(\psi) = steps(\psi')$, properties of ordinals yield $steps(\phi) = steps(\phi')$. We conclude by observing that for any suitable α, $\phi[\alpha] = (\psi \cdot \phi)[steps(\psi) + \alpha] = (\psi' \cdot \phi')[steps(\psi') + \alpha] = \phi'[\alpha]$. □

Proposition 5.4.28. *Let ψ, ϕ be stepwise-or-nil proof terms such that $\psi \approx_{()} \phi$. Then $\psi \equiv \phi$.*

Proof. We proceed by induction on α where $\psi \overset{\alpha}{\approx}_{()} \phi$. We analyse the rule used in the last step of the rebracketing equivalence derivation.

For the rules Refl, Symm and Trans, the result holds immediately.

Assume that the last used rule in the derivation is Eqn, so that $\psi = (\psi_1 \cdot \psi_2) \cdot \psi_3$ and $\phi = \psi_1 \cdot (\psi_2 \cdot \psi_3)$. In this case we can obtain $steps(\psi) = steps(\phi) > 0$ immediately. Let $\gamma < steps(\psi)$. If $\gamma < steps(\psi_1)$, then $\psi[\gamma] = (\psi_1 \cdot \psi_2)[\gamma] = \psi_1[\gamma] = \phi[\gamma]$. The other cases, i.e. $steps(\psi_1) \leqslant \gamma < steps(\psi_1) + steps(\psi_2)$ and $steps(\psi_1) + steps(\psi_2) \leqslant \gamma$, admit analogous arguments.

Assume that the last used rule is InfComp, so that $\psi = \cdot_{i<\omega} \psi_i$, $\phi = \cdot_{i<\omega} \phi_i$, and $\psi_n \overset{\beta_n}{\approx}_{()} \phi_n$ where $\beta_n < \alpha$, for all $n < \omega$. Then IH on each β_n implies $\psi_n \equiv \phi_n$. Therefore we obtain $steps(\psi) = steps(\phi) > 0$ immediately. To conclude it is enough to observe, for any $\gamma < steps(\psi)$, that Lem. 5.1.2 implies $\gamma = \sum_{i<n} steps(\psi_i) + \gamma_0$ where $\gamma_0 < steps(\psi_n)$, then (given IH on each $\psi_i \overset{\beta_i}{\approx}_{()} \phi_i$) $\psi[\gamma] = \psi_n[\gamma_0] = \phi_n[\gamma_0] = \phi[\gamma]$.

If the last used rule is Comp, then a similar argument applies.

Assume that the rule used in the last derivation step is Lim. Assume for contradiction $steps(\phi) > steps(\psi)$, so that the step $\phi[steps(\psi)]$ exists. Consider $k := max(mind(\phi[0]), mind(\phi[steps(\psi)]))$. Then there exist χ_k, ϕ'_k, ψ'_k verifying $\phi \overset{\alpha_k}{\approx}_{(B)} \chi_k \cdot \phi'_k$, $\psi \overset{\beta_k}{\approx}_{(B)} \chi_k \cdot \psi'_k$, $mind(\phi'_k) > k \geqslant mind(\phi[steps(\psi)])$, $mind(\psi'_k) > k$, $\alpha > \alpha_k$, and $\alpha > \beta_k$. Recalling that $\overset{\gamma}{\approx}_{(B)} \subseteq \overset{\gamma}{\approx}_{()}$ for any γ, we can apply IH to α_k obtaining $\phi \equiv \chi_k \cdot \phi'_k$, so that $\phi[steps(\psi)] = (\chi_k \cdot \phi'_k)[steps(\psi)]$. Therefore, assuming $steps(\psi) = steps(\chi_k) + \gamma$ would imply $\phi'_k[\gamma] = \phi[steps(\psi)]$ contradicting $mind(\phi'_k) > mind(\phi[steps(\psi)])$; cfr. Lem. 5.4.13. Then $steps(\psi) < steps(\chi_k)$. On the other hand, IH can be applied also to β_k, yielding $\psi \equiv \chi_k \cdot \psi'_k$, and therefore $steps(\psi) \geqslant steps(\chi_k)$, i.e. a contradiction. Consequently $steps(\phi) \leqslant steps(\psi)$. A similar argument yields $steps(\psi) \leqslant steps(\phi)$. Thus $steps(\psi) = steps(\phi)$.

Let $\gamma < steps(\psi)$. Then there exists χ, ψ', ϕ' such that $\psi \overset{\alpha_0}{\approx}_{(B)} \chi \cdot \psi'$, $\phi \overset{\beta_0}{\approx}_{(B)} \chi \cdot \phi'$, $mind(\psi') > mind(\psi[\gamma])$, $mind(\phi') > mind(\psi[\gamma])$, $\alpha_0 < \alpha$ and $\beta_0 < \alpha$. Then IH on α_0 and β_0 yields $\psi \equiv \chi \cdot \psi'$ and $\phi \equiv \chi \cdot \phi'$, so that $\psi[\gamma] = (\chi \cdot \psi')[\gamma]$ and $\phi[\gamma] = (\chi \cdot \phi')[\gamma]$. Observing that $\gamma = steps(\chi) + \gamma_0$ would imply $\psi[\gamma] = \psi'[\gamma_0]$, and then $mind(\psi') \leqslant mind(\psi[\gamma])$ (cfr. Lem. 5.4.13) thus producing a contradiction, we obtain $\gamma < steps(\chi)$. Then $\psi[\gamma] = \chi[\gamma]$, and also $\phi[\gamma] = \chi[\gamma]$. Hence $\psi[\gamma] = \phi[\gamma]$. □

Proposition 5.4.29. *Let ψ, ϕ such that $\psi \equiv \phi$. Then $\psi \approx_{()} \phi$.*

Proof. We proceed by induction on $\langle \alpha, \beta \rangle$ such that $\psi \in \mathbf{PT}_\alpha$ and $\phi \in \mathbf{PT}_\beta$.

If $\psi \in Ter^\infty(\Sigma)$, so that $steps(\psi) = 0$, then $\psi \equiv \phi$ implies $\psi = \phi$, hence we conclude immediately.

If ψ is a one-step, so that $steps(\psi) = 1$, then $\psi \equiv \phi$ implies $\psi = \psi[0] = \phi[0] = \phi$.

Assume $\psi = \psi_1 \cdot \psi_2$ and that it is not an infinite concatenation. In this case, $steps(\psi) = steps(\phi) > 1$ is a successor ordinal, so that $\phi = \phi_1 \cdot \phi_2$ and it is neither an infinite concatenation; cfr. Lem. 5.4.24. Observe that $\alpha = \alpha_1 + \alpha_2 + 1$ and $\beta = \beta_1 + \beta_2 + 1$, where $\psi_i \in \mathbf{PT}_{\alpha_i}$ and $\phi_i \in \mathbf{PT}_{\beta_i}$ for $i = 1, 2$. We analyse the different cases arising from the comparison between $steps(\psi_1)$ and $steps(\phi_1)$.

- Assume $steps(\psi_1) < steps(\phi_1)$. In this case we apply Lem. 5.4.25, obtaining that $\phi_1 \approx_{()} \chi_1 \cdot \chi_2$ and $steps(\chi_1) = steps(\psi_1)$ for some stepwise proof terms $\chi_1 \in \mathbf{PT}_{\gamma_1}$ and $\chi_2 \in \mathbf{PT}_{\gamma_2}$, and moreover, that $\gamma_1 < \beta_1$ and $\gamma_2 \leqslant \beta_1$.

  Therefore $\phi \approx_{()} (\chi_1 \cdot \chi_2) \cdot \phi_2 \approx_{()} \chi_1 \cdot (\chi_2 \cdot \phi_2)$, and hence Prop. 5.4.28 and hypotheses yield $\psi = \psi_1 \cdot \psi_2 \equiv \chi_1 \cdot (\chi_2 \cdot \phi_2) \equiv \phi$. Observe that for any $\beta < steps(\psi_1)$, $\psi_1[\beta] = \psi[\beta] = \phi[\beta] = (\chi_1 \cdot (\chi_2 \cdot \phi_2))[\beta] = \chi_1[\beta]$; consequently, $\psi_1 \equiv \chi_1$. In turn, Lem. 5.4.27 yields $\psi_2 \equiv \chi_2 \cdot \phi_2$.

  Observing that $\alpha_i < \alpha$ for $i = 1, 2$ suffices to enable the application of IH to both $\psi_1 \equiv \chi_1$ and $\psi_2 \equiv \chi_2 \cdot \phi_2$. Therefore, we conclude by Comp, Symm and Trans.

- Assume $steps(\psi_1) > steps(\phi_1)$. In this case, an analysis similar to that of the previous case yields $\psi_1 \approx_{(B)} \chi_1 \cdot \chi_2$ such that $steps(\chi_1) = steps(\phi_1)$, $\gamma_1 < \alpha_1$ and $\gamma_2 \leqslant \alpha_1$ where $\chi_i \in \mathbf{PT}_{\gamma_i}$ for $i = 1, 2$; therefore $\chi_1 \cdot (\chi_2 \cdot \psi_2) \equiv \psi \equiv \phi = \phi_1 \cdot \phi_2$; and consequently $\chi_1 \equiv \phi_1$ and $\chi_2 \cdot \psi_2 \equiv \phi_2$.

Observe $\gamma_1 < \alpha_1 < \alpha$. On the other hand, $\chi_2 \cdot \psi_2 \in \mathbf{PT}_\delta$ where $\delta = \gamma_2 + \alpha_2 + 1 \leqslant \alpha_1 + \alpha_2 + 1 = \alpha$, and $\beta_2 < \beta$. Therefore, IH can be applied to both $\chi_1 \equiv \phi_1$ and $\chi_2 \cdot \psi_2 \equiv \phi_2$, so that we conclude as in the previous case.

- Assume $steps(\psi_1) = steps(\phi_1)$. Then a simple analysis of the components of ψ_1 and ϕ_1 yields $\psi_1 \equiv \phi_1$. In turn, this assertion allows to apply Lem. 5.4.27 to obtain $\psi_2 \equiv \phi_2$. Applying IH to both ψ_i we obtain $\psi_1 \approx_{()} \phi_1$ and $\psi_2 \approx_{()} \phi_2$. Hence we conclude by Comp.

Assume $\psi = \cdot_{i<\omega}\ \psi_i$. In this case, a simple argument based on Lem. 5.4.24 yields $\phi = \cdot_{i<\omega}\ \phi_i$.

As the verification for this case involves a great number of technical details, we describe the idea first. We define a stepwise proof term $\chi = \cdot_{i<\omega}\ \chi_i$ enjoying the following properties: $\psi \approx_{()} \chi$, and $\chi_n \equiv \phi_n$ for all $n < \omega$. The Lim rule is used in the last step of the derivation $\psi \approx_{()} \chi$, verifying that the corresponding premises are valid w.r.t. $\approx_{(B)}$. In turn, Lem. 5.4.23 allows to apply IH on any χ_n, since $\chi \in \mathbf{PT}_\delta$ implies $\delta = steps(\chi) = steps(\psi) = \alpha$ (cfr. Prop. 5.4.28). Therefore we obtain $\chi_n \approx_{()} \phi_n$ for all $n < \omega$, implying $\chi \approx_{()} \phi$. Then Trans yields $\psi \approx_{()} \phi$. A very schematic derivation tree follows:

$$\dfrac{\dfrac{\cdots \quad \begin{array}{c}\psi \approx_{(B)} \xi_k \cdot \psi' \\ \chi \approx_{(B)} \xi_k \cdot \chi'\end{array} \quad \cdots}{\psi \approx_{()} \chi}\,\mathsf{Lim} \qquad \dfrac{\cdots \quad \dfrac{B_n}{\chi_n \approx_{()} \phi_n} \quad \cdots}{\chi \approx_{()} \phi}\,\mathsf{InfComp}}{\psi \approx_{()} \phi}\,\mathsf{Trans}$$

where we can observe the soundness of the derivation, even if Lim is applied in some of the B_n derivations.

We define χ_k, by induction on k, for all $k < \omega$. We observe that $\sum_{i<k} steps(\phi_i) < steps(\phi) = steps(\psi)$. Then we define, along with χ_k, two values p_k and β_k as follows: $p_0 := 0$, $\beta_0 := 0$, and if $k > 0$, then p_k and β_k are the unique (cfr. Lem. 5.1.2) values verifying $\sum_{i<k} steps(\phi_i) = \sum_{i<p_k} steps(\psi_i) + \beta_k$ and $\beta_k < steps(\psi_{p_k})$. We also define $p' := p_{k+1} - 1$. Simultaneously with the definition of χ_k, we will verify the following auxiliary assertion:

- $\chi_0 \cdot \ldots \cdot \chi_k \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'}$ if $\beta_{k+1} = 0$; and
- there exist χ', ξ such that $\psi_{p_{k+1}} \approx_{(B)} \chi' \cdot \xi$, $steps(\chi') = \beta_{k+1}$ and $\chi_0 \cdot \ldots \cdot \chi_k \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'} \cdot \chi'$ (or $\chi_0 \cdot \ldots \cdot \chi_k \approx_{(B)} \chi'$ if $p_{k+1} = 0$), if $\beta_{k+1} > 0$.

Therefore, when defining χ_n for a given n, we can consider this assertion to be valid for all $n' < n$.

Let $n < \omega$. Several cases must be analysed to define χ_n.

- Assume that either $n = 0$, i.e. the base case, or $n > 0$ and $\beta_n = 0$.
  - Assume $p_n = p_{n+1}$, implying $steps(\phi_n) = \beta_{n+1} > 0$, so that $steps(\phi_n) < steps(\psi_{p_n})$. In this case we define χ_n to be some term verifying $\psi_{p_n} \approx_{(B)} \chi_n \cdot \xi$ and $steps(\chi_n) = steps(\phi_n)$; cfr. Lem. 5.4.25.
  - Assume $p_n < p_{n+1}$ and $\beta_{n+1} = 0$, so that $steps(\phi_n) = steps(\psi_{p_n}) + \ldots + steps(\psi_{p'})$. In this case we define $\chi_n := \psi_{p_n} \cdot \ldots \cdot \psi_{p'}$.
  - Assume $p_n < p_{n+1}$ and $\beta_{n+1} > 0$, implying $steps(\phi_n) = steps(\psi_{p_n}) + \ldots + steps(\psi_{p'}) + \beta_{n+1}$. We consider some χ', ξ verifying $\psi_{p_{n+1}} \approx_{(B)} \chi' \cdot \xi$ and $steps(\chi') = \beta_{n+1}$; cfr. Lem. 5.4.25. Then we define $\chi_n := \psi_{p_n} \cdot \ldots \cdot \psi_{p'} \cdot \chi'$.

In any case, if $n = 0$ then the auxiliary assertion holds immediately; otherwise, it suffices to apply the same assertion on $n-1$ obtaining $\chi_0 \cdot \ldots \cdot \chi_{n-1} \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p_n-1}$, and then Refl and Comp.

- Assume $\beta_n > 0$. In this case $n > 0$, then the auxiliary assertion on $n-1$ implies the existence of χ', ξ verifying $\psi_{p_n} \approx_{(B)} \chi' \cdot \xi$, $steps(\chi') = \beta_n$ and $\chi_0 \cdot \ldots \cdot \chi_{n-1} \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p_n-1} \cdot \chi'$ (or $\chi_0 \cdot \ldots \cdot \chi_{n-1} \approx_{(B)} \chi'$ if $p_n = 0$).
  - Assume $p_{n+1} = p_n$, implying $\beta_{n+1} = \beta_n + steps(\phi_n) < steps(\psi_{p_n}) = \beta_n + steps(\xi)$, implying $steps(\phi_n) < steps(\xi)$. In this case we define χ_n to be some term verifying $\xi \approx_{(B)} \chi_n \cdot \xi'$ and $steps(\chi_n) = steps(\phi_n)$; cfr. Lem. 5.4.25. Observe $\psi_{p_n} \approx_{(B)} (\chi' \cdot \chi_n) \cdot \xi'$, $steps(\chi' \cdot \chi_n) = \beta_n + steps(\phi_n) = \beta_{n+1}$ and $\chi_0 \cdot \ldots \cdot \chi_n \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p_n-1} \cdot (\chi' \cdot \chi_n)$, then the auxiliary statement holds for n; recall $\beta_{n+1} > \beta_n \geqslant 0$.
  - Assume $p_{n+1} = p_n + 1$ and $\beta_{n+1} = 0$, implying $steps(\psi_{p_n}) = \beta_n + steps(\phi_n)$. Observe $steps(\xi) = steps(\phi_n)$. We define $\chi_n := \xi$. Then $\chi_0 \cdot \ldots \chi_n \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p_n-1} \cdot \chi' \cdot \xi \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p_n-1} \cdot \psi_{p_n}$, then the auxiliary statement holds for n.
  - Assume $p_{n+1} > p_n+1$ and $\beta_{n+1} = 0$, implying $steps(\phi_n) = \beta' + steps(\psi_{p_n+1}) + \ldots + steps(\psi_{p'})$, where $steps(\psi_{p_n}) = \beta_n + \beta'$. Observe $steps(\xi) = \beta'$. We define $\chi_n := \xi \cdot \psi_{p_n+1} \cdot \ldots \cdot \psi_{p'}$. We verify the auxiliary statement for n similarly to the previous case.
  - Assume $p_{n+1} > p_n$ and $\beta_{n+1} > 0$, implying $steps(\phi_n) = \beta' + steps(\psi_{p_n+1}) + \ldots + steps(\psi_{p'}) + \beta_{n+1}$ (or just $\beta' + \beta_{n+1}$ if $p_{n+1} = p_n + 1$), where $steps(\psi_{p_n}) = \beta_n + \beta'$. Observe $steps(\xi) = \beta'$. Let χ'', ξ' such that $\psi_{p_{n+1}} \approx_{(B)} \chi'' \cdot \xi'$ and $steps(\chi'') = \beta_{n+1}$. We define $\chi_n := \xi \cdot \psi_{p_n+1} \cdot \ldots \cdot \psi_{p'} \cdot \chi''$ (or just $\xi \cdot \chi''$ if $p_{n+1} = p_n + 1$). We verify the auxiliary statement for n similarly to the previous cases.

In turn, a simple analysis of each case yields $steps(\chi_n) = steps(\phi_n)$ for each $n < \omega$.

We verify $\psi \equiv \chi$, since this assertion is used when obtaining $\psi \approx_{()} \chi$. Given $steps(\chi_n) = steps(\phi_n)$ for all $n < \omega$, we obtain immediately $steps(\chi) = steps(\phi) = steps(\psi)$ (recall the hypothesis $\psi \equiv \phi$). Let $\beta < steps(\chi)$, let n be a natural number verifying $\beta < \sum_{i \leqslant n} steps(\chi_i)$ (cfr. Lem 5.1.2). Then $\chi[\beta] = (\chi_0 \cdot \ldots \cdot \chi_n)[\beta]$. Observe $\chi_0 \cdot \ldots \cdot \chi_n \approx_{(B)} \psi'$ for some ψ' verifying $\psi \approx_{(B)} \psi' \cdot \psi''$, cfr. the auxiliary assertion in the definition of χ_n, so that $steps(\psi') = \sum_{i \leqslant n} steps(\chi_i) > \beta$. Therefore $\chi[\beta] = (\chi_0 \cdot \ldots \cdot \chi_n)[\beta] = \psi'[\beta] = \psi[\beta]$, cfr. Prop. 5.4.28. Hence $\psi \equiv \chi$.

We verify $\psi \approx_{()} \chi$. Let $k < \omega$, let p such that $p > 0$ and $mind(\psi_i) > k$ if $i > p$. Let n be a natural number verifying $\sum_{i \leqslant n} steps(\phi_i) > \sum_{i \leqslant p} steps(\psi_i)$. Observe that $p_{n+1} > p$. We analyse the two possible cases of the auxiliary statement in the definition of χ_n; again, $p' := p_{n+1} - 1$.

If $\beta_{n+1} = 0$, then $\chi_0 \cdot \ldots \cdot \chi_n \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'}$; observe that also $\chi \approx_{(B)} \chi_0 \cdot \ldots \cdot \chi_n \cdot (\cdot_{i<\omega}\, \chi_{n+1+i})$ and $\psi \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'} \cdot (\cdot_{i<\omega}\, \psi_{p_{n+1}+i})$. We obtain immediately $\chi \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'} \cdot (\cdot_{i<\omega}\, \chi_{n+1+i})$ and $mind(\cdot_{i<\omega}\, \psi_{p_{n+1}+i}) > k$, since $p_{n+1} > p$. Prop. 5.4.28 yields $\chi_0 \cdot \ldots \cdot \chi_n \equiv \psi_0 \cdot \ldots \cdot \psi_{p'}$, so that Lem. 5.4.27 can be applied to obtain $\cdot_{i<\omega}\, \chi_{n+1+i} \equiv \cdot_{i<\omega}\, \psi_{p_{n+1}+i}$, and therefore $mind(\cdot_{i<\omega}\, \chi_{n+1+i}) > k$, cfr. Lem. 5.4.13.

Otherwise, there exist some χ', ξ such that $\chi_0 \cdot \ldots \cdot \chi_n \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'} \cdot \chi'$ and $\psi_{p_{n+1}} \approx_{(B)} \chi' \cdot \xi$. By an argument analogous to that of the previous case, we obtain

$\psi \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'} \cdot \chi' \cdot (\xi \cdot \cdot_{i<\omega} \psi_{p_{n+1}+1+i})$, $\chi \approx_{(B)} \psi_0 \cdot \ldots \cdot \psi_{p'} \cdot \chi' \cdot (\cdot_{i<\omega} \chi_{n+1+i})$, and $mind(\xi \cdot \cdot_{i<\omega} \psi_{p_{n+1}+1+i}) = mind(\cdot_{i<\omega} \chi_{n+1+i}) > k$.

Consequently we can apply Lim to obtain $\psi \approx_{()} \chi$. Observe that the premises of the Lim application correspond to the $\approx_{(B)}$ relation, so that the derivation is sound.

The only element needed to complete the idea described earlier, and then to conclude the proof, is to obtain $\chi_n \equiv \phi_n$ for all n. We have already obtained $\psi \equiv \chi$, so that the hypothesis $\psi \equiv \phi$ implies $\chi \equiv \phi$. On the other hand, we have also obtained $steps(\chi_n) = steps(\phi_n)$ for all n. Then a simple induction on n yields $\chi_n \equiv \phi_n$ for all n. Thus we conclude. □

Theorem 5.4.30. *Let ψ, ϕ be stepwise-or-nil proof terms. Then $\psi \approx_{()} \phi$ iff $\psi \equiv \phi$.*

Proof. Immediate corollary of Prop. 5.4.28 and Prop. 5.4.29. □

5.5 Compression

The compression lemma, [KKSdV90, KKSdV95, BKdV03, Ket12], establishes that the full power of left-linear, strongly convergent reduction can be achieved considering only reduction sequences having length at most ω, i.e. the first infinite ordinal. Formally, the lemma states that for any strongly convergent reduction sequence $t \xrightarrow{\delta}\!\!\!\!\twoheadrightarrow u$ in a left-linear TRS, there exists another strongly convergent reduction sequence $t \xrightarrow{\gamma}\!\!\!\!\twoheadrightarrow u$ such that $\mathsf{length}(\gamma) \leqslant \omega$. In [KKSdV95] a more precise statement is given: for orthogonal TRSs, the reduction sequence γ can be chosen such that it is Lévy-equivalent (cfr. [HL91]) to δ. Cfr. Section 1.2.3, where a reduction sequence which length is $\omega \times 2$, and its compressed version having length ω, are presented. We point out that the compression result is not valid in general, if we consider the *weak convergence* criterion for the definition of infinitary reduction sequences, instead of the strong convergence criterion we use in this thesis; cfr. [KKSdV95] p. 22.[12] The compression result is also invalid, in the general case, for *non-left-linear*, first-order term rewriting systems; an analysis of compression for those systems can be found in [Ket12].

The aim of this section is to present a novel proof of the property of compression for convergent, left-linear, first-order term rewriting, based on the characterisation of permutation equivalence given in Section 5.3. Given that any convergent reduction sequence can be described by means of a proof term, cfr. Prop. 5.4.19, compression can be studied within the framework given by proof terms. In this setting, the compression result can be stated as follows: for any convergent proof term ψ, there exists a stepwise-or-nil proof term (cfr. Dfn. 5.4.2) ϕ such that $\psi \approx \phi$ and $steps(\phi) \leqslant \omega$. As observed in the introduction to this chapter, the obtained result is more general than the statements present in the referenced literature, in two ways. First, the result applies to orthogonal *reduction sequences*, even for non-orthogonal TRSs, while at the same time, the equivalence between the original and the compressed contraction activities is asserted. Secondly, the result applies to (the description of) arbitrary contraction activities, that is, it is not limited to reduction sequences. Put in this way, the compression result indicates that any orthogonal contraction activity can be performed in a sequential fashion, involving at most ω steps.

[12] The weak and strong convergence approaches to infinitary rewriting are discussed in Section 5.1.4, prior to the definition of reduction sequence, Dfn. 5.1.32.

This proof resorts to a key technical result, namely the ability of **factorising** (more precisely, obtaining a factorised version of) any proof term, in a leading part denoting *finite* contraction activity, followed by a tail denoting activity at *arbitrary depths*; cfr. the notion of *minimum activity depth*, formalised in Dfn. 5.1.38, Dfn. 5.2.6 and Dfn. 5.2.8. The characterisation of permutation equivalence shows that the original proof term and its factorised version denote the contraction of the same steps, while the concatenation symbol allows to separate the leading part from the tail in the factorised version. Formally, the main auxiliary result for the compression proof is the existence, for any proof term ψ and $n < \omega$, of two proof terms χ and ϕ, such that $\psi \approx_B \chi \cdot \phi$, χ is a finite stepwise-or-nil proof term, and $mind(\phi) > n$.

In the following, we develop the technical work leading to the factorisation result. Subsequently, we give a statement of the compression lemma based in proof terms and permutation equivalence, and prove it by resorting to factorisation.

5.5.1 Factorisation for infinitary multisteps

In this section, a factorisation result for the particular case of infinitary multisteps is stated an proved. The proof is based on the concept of **collapsing sequence of positions** for an infinitary multistep. Such a sequence indicates that the contraction activity denoted by the infinitary multistep includes a series of reduction steps which can be performed consecutively and at the same position, so that all of these steps, except possibly the last one, correspond to collapsing rules.

I.e., considering the rules $\mu : f(x) \to g(x)$, $\rho : i(x) \to x$ and $\rho' : j(x) \to x$, the proof term $h(\rho(\rho'(\mu(a))), \mu(b))$ includes a finite collapsing sequence formed by the occurrences of ρ and ρ' plus the leftmost occurrence of μ. This collapsing sequence indicates that a sequentialisation of the activity denoted by this proof term can include up to three consecutive collapsing steps at the same position.

On the other hand, the proof term ρ^ω includes an *infinite* collapsing sequence. Observe that this proof term is *not convergent*. In the following, a relation between infinite collapsing sequences and non-convergence is shown [13], and later exploited in the proof of the factorisation result for infinitary multisteps.

Definition 5.5.1. *Let ψ be an infinitary multistep. A sequence $\langle p_i \rangle_{i \leqslant n}$ (resp. $\langle p_i \rangle_{i < \omega}$) is a* finite *(resp.* infinite*)* collapsing sequence *for ψ iff for all $i < n$ (resp. $i < \omega$), $\psi(p_i) = \mu$ where $\mu : l[x_1, \ldots, x_m] \to x_j$ and $p_{i+1} = p_i\, j$.*

Observe that the length of $\langle p_i \rangle_{i \leqslant n}$ is $n + 1$. Moreover, for any $\langle p_i \rangle_{i \leqslant n}$ or $\langle p_i \rangle_{i < \omega}$, an easy induction (on $k - j$) yields that $j < k < \omega$ implies $p_j < p_k$.

[13]We conjecture that, in fact, non-convergence of infinitary multisteps, and therefore non-termination of developments of orthogonal sets of redex occurrences in first-order rewriting, can be fully characterised by means of collapsing sequences. This observation suggests that infinitary multisteps could be used as a technical tool to study termination of developments in infinitary rewriting, leading to an approach being alternative to e.g. that described in [BKdV03], Sec. 12.5. In this work, only the material needed for the factorisation result is developed. Some conjectures follow; further investigation about this subject is left as future work.

Observe that infinitary multisteps exist being tgt_T-WN^∞ and including infinite collapsing sequences. E.g., if we add the rule $\tau : h(x, y) \to y$, then $\tau(\rho^\omega, a)$ has a as tgt_T-normal form. Intuitively, including a collapsing sequence implies that an infinitary multistep is not tgt_T-WN^∞, only if that collapsing sequence cannot be erased. Then we state the following conjecture: an infinitary multistep is tgt_T-WN^∞ iff it does not include any infinite collapsing sequence at a *non-erasable* position, where a position p is erasable for ψ iff $p = p_1 i p_2$, $\psi(p_1) = \mu$, and the i-th variable in the left-hand side of μ does not occur in the corresponding right-hand side.

Lemma 5.5.2. *Let ψ be a proof term, $\langle p_i \rangle_{i \leqslant n}$ (resp. $\langle p_i \rangle_{i<\omega}$) a collapsing sequence for ψ, and j, k such that $j + k \leqslant n$ (resp $j, k < \omega$). Then $\langle p_{j+i} \rangle_{i \leqslant k}$ is a collapsing sequence for ψ.*

Proof. Easy consequence of Dfn. 5.5.1. □

Notice that Lem. 5.5.2 implies particularly that $\langle p_i \rangle_{i \leqslant k}$ is a collapsing sequence if $k \leqslant n$ (resp. $k < \omega$).

For any ψ infinitary multistep and $p \in \texttt{Pos}(\psi)$, we observe that $\langle p \rangle$ is a collapsing sequence for ψ whose length is 1. This is an easy *existence* result. A *uniqueness* result for collapsing sequences holds as well, namely:

Lemma 5.5.3. *Let ψ be an infinitary multistep, $p \in \texttt{Pos}(\psi)$, and n such that $0 < n < \omega$. Then there is at most one collapsing sequence for ψ starting at p and having length n.*

Proof. We proceed by induction on n. If $n = 1$ then the result holds immediately since the only suitable sequence is $\langle p \rangle$.

Let $n = n' + 1$. Let $\langle p_i \rangle_{i \leqslant n'}$ and $\langle q_i \rangle_{i \leqslant n'}$ two collapsing sequences for ψ, both starting with p. Lem. 5.5.2 implies that both $\langle p_i \rangle_{i \leqslant (n'-1)}$ and $\langle q_i \rangle_{i \leqslant (n'-1)}$ are collapsing sequences for ψ. Then IH on n' implies $p_i = q_i$ if $i < n'$, so that particularly $p_{n'-1} = q_{n'-1}$. Applying Dfn. 5.5.1 on $\langle p_i \rangle_{i \leqslant n'}$ and $\langle q_i \rangle_{i \leqslant n'}$ yields $\psi(p_{n'-1}) = \psi(q_{n'-1}) = \mu$ such that $\mu : l[x_1, \ldots, x_m] \to x_j$ and $p_{n'} = q_{n'} = p_{n'-1}\, j$. Thus we conclude. □

Lemma 5.5.4. *Let ψ be an infinitary multistep, $p \in \texttt{Pos}(\psi)$, and $n, k < \omega$ (resp $n < \omega$), such that both $\langle p_i \rangle_{i \leqslant n}$ and $\langle q_i \rangle_{i \leqslant n+k}$ (resp., and $\langle q_i \rangle_{i<\omega}$) are collapsing sequences for ψ starting with p. Then $i \leqslant n$ implies $q_i = p_i$.*

Proof. Easy consequence of Lem. 5.5.2 and Lem. 5.5.3. □

We already remarked that any prefix of an infinite collapsing sequence is a collapsing sequence as well. Conversely, a sequence of growing collapsing sequences starting at the same position indicates the presence of an infinite collapsing sequence. The following lemma formalises this idea.

Lemma 5.5.5. *Let ψ be an infinitary multistep and $p \in \texttt{Pos}(\psi)$, such that for any $n < \omega$, there is a collapsing sequence for ψ starting at p and having length n. Then there is an infinite collapsing sequence for ψ starting at p.*

Proof. We define the sequence $\langle p_i \rangle_{i<\omega}$ as follows: for all $k < \omega$, $p_k := q_k$ where $\langle q_i \rangle_{i \leqslant k}$ is the only (cfr. Lem. 5.5.3) collapsing sequence for ψ starting at p and having length $k+1$. Let $j < \omega$, and $\langle q_i \rangle_{i \leqslant j}$ and $\langle q'_i \rangle_{i \leqslant (j+1)}$ the collapsing sequences for ψ starting at p and having lengths $j+1$ and $j+2$ respectively. Observe that Lem. 5.5.4 implies $p_j = q_j = q'_j$; on the other hand, $p_{j+1} = q'_{j+1}$. Then $\langle q'_i \rangle_{i \leqslant (j+1)}$ being a collapsing sequence implies that $\psi(p_j) = \psi(q'_j) = \mu$ where $\mu : l[x_1, \ldots, x_m] \to x_i$ and $p_{j+1} = q'_{j+1} = q'_j\, i = p_j\, i$. Consequently, $\langle p_i \rangle_{i<\omega}$ is a collapsing sequence. Thus we conclude. □

After this general presentation of collapsing sequences, we will focus on collapsing sequences starting with ϵ. The existence of an infinite collapsing sequence starting with ϵ is invariant w.r.t. partial computation of the target of an infinitary multistep. This implies that an infinitary multistep including such a sequence is non-convergent, i.e. its target cannot be computed, cfr. Dfn. 5.2.4 and Dfn. 5.2.5.

Lemma 5.5.6. *Let ψ be an infinitary multistep, $\langle p_i \rangle_{i<\omega}$ a collapsing sequence for ψ starting at ϵ, and $\psi \xrightarrow[tgt_T]{\delta} \phi$. Then there exists some $\langle q_i \rangle_{i<\omega}$ being a collapsing sequence for ϕ starting at ϵ.*

Proof. We proceed by transfinite induction on $\mathsf{length}(\delta)$. If $\mathsf{length}(\delta) = 0$, so that $\phi = \psi$, then we conclude immediately.

Assume $\mathsf{length}(\delta) = \alpha + 1$, so that $\psi \xrightarrow[tgt_T]{\delta'} \chi \xrightarrow[tgt_T]{a} \phi$ where $\mathsf{length}(\delta') = \alpha$; let us say $a = \langle \chi, r, \underline{\mu}, \sigma \rangle$, and define $d := d(a) = |r|$, where $\underline{\mu}$ is the rule in tgt_T corresponding to a rule μ in the object TRS. IH can be applied on δ', obtaining the existence of $\langle p'_i \rangle_{i<\omega}$, a collapsing sequence for χ starting at ϵ. Observe that $\phi = \chi[\sigma h]_r$ where $\mu : l[x_1, \ldots, x_m] \to h$, so that $\underline{\mu} : \mu(x_1, \ldots, x_m) \to h$, implying $\sigma = \{x_i := \chi|_{r\,i}\}$. Notice also that $|p'_n| = n$ for all $n < \omega$, implying $|p'_d| = |r|$. We consider two cases.

- Assume $p'_d \parallel r$. Let $n < \omega$. Observe that $n < d$, resp. $n > d$, implies $p'_n < p'_d$, resp. $p'_d < p'_n$. In either case, $r \leqslant p'_n$ would contradict $p'_d \parallel r$, in the former case by transitivity of $<$, in the latter since all prefixes of p'_n form a total order in a tree domain. Hence $r \nleqslant p'_n$. Consequently, for all $n < \omega$, $p'_n \in \mathtt{Pos}(\phi)$ and $\phi(p'_n) = \chi(p'_n)$. Thus $\langle p'_n \rangle_{n<\omega}$ is a collapsing sequence for ϕ.

- Assume $p'_d = r$. In this case, $\mu : l[x_1, \ldots, x_m] \to x_j$ and $p'_{d+1} = p'_d\, j$, so that $\phi = \chi[\chi|_{p'_{d+1}}]_{p'_d}$. Observe that for any position p'', $\phi|_{p'_d\, p''} = \chi|_{p'_{d+1}\, p''}$.

  Let $\langle q_i \rangle_{i<\omega}$ be the sequence defined as follows:
  $$q_n := \begin{cases} p'_n & \text{if } n \leqslant d \\ p'_d p'' \text{ where } p'_{n+1} = p'_{d+1} p'' & \text{if } n > d \end{cases}$$
  Let $n < \omega$. If $n < d$, then $q_n = p'_n < p'_d$, so that $\phi(q_n) = \phi(p'_n) = \chi(p'_n) = \nu$ where $\nu : l[y_1, \ldots, y_m] \to y_i$ and $q_{n+1} = p'_{n+1} = p'_n\, i = q_n\, i$. Now assume $n \geqslant d$. Let p'' such that $p'_{n+1} = p'_{d+1} p''$, observe that $n = d$ implies $p'' = \epsilon$. Observe $\chi(p'_{n+1}) = \nu$, $\nu : l[y_1, \ldots, y_m] \to y_i$ and $p'_{n+2} = p'_{n+1}\, i = p'_{d+1} p''\, i$. On the other hand, $q_n = p'_d p''$ (if $n = d$, then $q_n = p'_d = p'_d p''$ since in this case $p'' = \epsilon$), $q_{n+1} = p'_d p''\, i = q_n\, i$, and in turn $\phi(q_n) = \phi(p'_d p'') = \chi(p'_{d+1} p'') = \chi(p'_{n+1}) = \nu$.

  Hence $\langle q_i \rangle_{i<\omega}$ is a collapsing sequence for ϕ. Thus we conclude by observing that $q_0 = p'_0 = \epsilon$.

Assume that $\mathsf{length}(\delta)$ is a limit ordinal. For any $n < \omega$, we define β_n, χ_n, $\langle p^n_i \rangle_{i<\omega}$ and q_n as follows: β_n is an ordinal such that $\beta_n < \mathsf{length}(\delta)$ and $d(\delta[\gamma]) > n$ if $\beta_n \leqslant \gamma < \mathsf{length}(\delta)$; and χ_n is the infinitary multistep verifying $\psi \xrightarrow[tgt_T]{\delta[0,\beta_n)} \chi_n \xrightarrow[tgt_T]{\delta[\beta_n,\mathsf{length}(\delta))} \phi$. Observe that we can assume wlog that $\beta_n \leqslant \beta_{n+1}$. In turn, IH on $\delta[0, \beta_n)$ and Lem. 5.5.3 imply the existence of a unique collapsing sequence for χ_n starting at ϵ; we define $\langle p^n_i \rangle_{i<\omega}$ to be that sequence, and $q_n := p^n_n$.

Let $n < \omega$. Then Lem. 5.5.2 implies that $\langle p^n_i \rangle_{i\leqslant n}$ is a collapsing sequence for χ_n. Moreover, $\beta_n = \beta_{n+1}$ implies $\chi_n = \chi_{n+1}$, and otherwise $\beta_n < \beta_{n+1}$, so that $\psi \xrightarrow[tgt_T]{\delta[0,\beta_n)} \chi_n \xrightarrow[tgt_T]{\delta[\beta_n,\beta_{n+1})} \chi_{n+1}$ where $mind(\delta[\beta_n, \beta_{n+1})) > n$. Furthermore, $\chi_{n+1} \xrightarrow[tgt_T]{\delta[\beta_{n+1},\mathsf{length}(\delta))} \phi$ and $mind(\delta[\beta_{n+1}, \mathsf{length}(\delta))) > n$. Therefore $\mathtt{dist}(\chi_n, \chi_{n+1}) < 2^{-n}$ and $\mathtt{dist}(\chi_{n+1}, \phi) < 2^{-(n+1)}$ by Lem. 5.1.45; in turn Lem. 5.1.25 implies $\mathtt{dist}(\chi_n, \phi) < 2^{-n}$.

Then for any $j \leqslant n$, $\chi_n(p^n_j) = \chi_{n+1}(p^n_j) = \phi(p^n_j)$ since $|p^n_j| = j$. Therefore $\langle p^n_i \rangle_{i \leqslant n}$ is a collapsing sequence for χ_{n+1}, so that Lem. 5.5.3 implies $p^n_j = p^{n+1}_j$ if $j \leqslant n$. Hence $q_n = p^{n+1}_n$, so that $\phi(q_n) = \chi_{n+1}(q_n) = \nu$ where $\nu : l[x_1, \ldots, x_m] \to x_i$ and $q_{n+1} = p^{n+1}_{n+1} = p^{n+1}_n\, i = q_n\, i$. Consequently, $\langle q_i \rangle_{i<\omega}$ is a collapsing sequence for ϕ. Thus we conclude by observing $q_0 = \epsilon$. □

Lemma 5.5.7. *Let ψ be an infinitary multistep such that an infinite collapsing sequence for ψ starting at ϵ exists. Then ψ is not tgt_T-weakly normalising.*

Proof. Let $\psi \underset{tgt_T}{\twoheadrightarrow} \phi$. Then Lem. 5.5.6 implies that an infinite collapsing sequence for ϕ starting at ϵ exists, so that ϕ is not a tgt_T-normal form. Thus we conclude. □

On the other hand, the nonexistence of arbitrarily large collapsing sequences starting at ϵ allows a finite tgt_T-reduction sequence ending in a proof term having a function symbol at the root. In turn, for any finite tgt_T-reduction sequence there is a corresponding finite stepwise-or-nil proof term.

Lemma 5.5.8. *Let ψ be an infinitary multistep and n verifying $1 < n < \omega$, such that there is no collapsing sequence for ψ starting at ϵ and having length n. Then there exists a tgt_T-reduction sequence δ verifying $\psi \overset{\delta}{\underset{tgt_T}{\twoheadrightarrow}} \phi$, $\mathsf{length}(\delta) < n$, $d(\delta[i]) = 0$ for all $i < \mathsf{length}(\delta)$, and $\phi(\epsilon) \in \Sigma$.*

Proof. We proceed by induction on n.

Assume $n = 2$. If $\psi(\epsilon) \in \Sigma$ then we conclude immediately. Otherwise $\psi(\epsilon) = \mu$ where $\mu : l \to f(t_1, \ldots, t_k)$, so that the corresponding rule in tgt_T is $\underline{\mu} : \mu(x_1, \ldots, x_m) \to f(t_1, \ldots, t_k)$, and therefore $\psi \overset{(\epsilon,\underline{\mu})}{\underset{tgt_T}{\longrightarrow}} f(t'_1, \ldots, t'_k)$; thus we conclude by taking $\delta := \langle (\epsilon, \underline{\mu}) \rangle$.

Assume $n = n' + 1$ and $1 < n' < \omega$. If $\psi(\epsilon) \in \Sigma$ or $\psi(\epsilon) = \mu$, $\mu : l \to h$ and $h \notin \mathtt{Var}$, then the argument of the previous case allows to conclude. Otherwise, i.e. if $\psi(\epsilon) = \mu$ and $\mu : l[x_1, \ldots, x_m] \to x_k$, then the corresponding rule in tgt_T is $\underline{\mu} : \mu(x_1, \ldots, x_m) \to x_k$, implying that $\psi \overset{(\epsilon,\underline{\mu})}{\underset{tgt_T}{\longrightarrow}} \psi|_k$. Observe that $\langle p_i \rangle_{i \leqslant n'}$ being a collapsing sequence for $\psi|_k$ starting at ϵ would imply $(\langle \epsilon \rangle ; \langle k\, p_i \rangle_{i \leqslant n'})$ to be a collapsing sequence for ψ having length n, thus contradicting the lemma hypotheses. Indeed, if we define $\langle q_i \rangle_{i \leqslant n}$ as the given sequence for ψ, then $q_0 = \epsilon$ and $q_1 = k$, so that the condition on collapsing sequences holds for $j = 0$. If $0 < j < n$, then $q_j = k\, p_{j-1}$, so that $\psi(q_j) = \psi|_k\,(p_{j-1}) = \nu$ where $\nu : l[y_1, \ldots, y_m] \to y_i$ and $p_j = p_{j-1}\, i$, implying $q_{j+1} = k\, p_j = k\, p_{j-1}\, i = q_j\, i$.

Therefore IH can be applied to $\psi|_k$, yielding the existence of a reduction sequence δ' verifying $\psi|_k \overset{\delta'}{\underset{tgt_T}{\twoheadrightarrow}} \phi$, $\mathsf{length}(\delta') < n'$, $d(\delta'[i]) = 0$ for all $i < n'$, and $\phi(\epsilon) \in \Sigma$. Thus we conclude by taking $\delta := (\epsilon, \underline{\mu}); \delta'$. □

Lemma 5.5.9. *Let ψ be an infinitary multistep, and $\psi \overset{a}{\underset{tgt_T}{\longrightarrow}} \phi$. Then there exists a one-step χ such that $\psi \approx_B \chi \cdot \phi$ and $d(\chi) = d(a)$.*

Proof. We proceed by induction on $d(a)$.

Assume $a = (\epsilon, \underline{\mu})$, say $\mu : l[x_1, \ldots, x_m] \to h[x_1, \ldots, x_m]$ so that the corresponding rule in tgt_T is $\underline{\mu} : \mu(x_1, \ldots, x_m) \to h[x_1, \ldots, x_m]$. Therefore $\psi = \mu(\psi_1, \ldots, \psi_m)$ and

$\phi = h[\psi_1, \ldots, \psi_m]$. We take $\chi := \mu(src(\psi_1), \ldots, src(\psi_m))$. Then (OutIn) yields exactly $\psi \approx_B \chi \cdot \phi$. Thus we conclude.

Assume $a = (ip, \underline{\mu})$, so that $\psi = f(\psi_1, \ldots, \psi_i, \ldots, \psi_m)$, $\phi = f(\psi_1, \ldots, \phi_i, \ldots, \psi_m)$, and $\psi_i \xrightarrow[tgt_T]{(p,\underline{\mu})} \phi_i$. Then IH on $(p, \underline{\mu})$ implies $\psi_i \approx_B \chi_i \cdot \phi_i$ where χ_i is a one-step verifying $d(\chi_i) = |p|$. We take $\chi := f(src(\psi_1), \ldots, \chi_i, \ldots, src(\psi_m))$. Observe that for any $j \neq i$, (IdLeft) implies $\psi_j \approx_B src(\psi_j) \cdot \psi_j$, so that

$$\begin{array}{rcl} \psi & \approx_B & f(src(\psi_1) \cdot \psi_1, \ldots, \chi_i \cdot \phi_i, \ldots, src(\psi_m) \cdot \phi_m) \\ & \approx_B & f(src(\psi_1), \ldots, \chi_i, \ldots, src(\psi_m)) \cdot f(\psi_1, \ldots, \phi_i, \ldots, \psi_m) \\ & = & \chi \cdot \phi \end{array}$$

Thus we conclude by noticing that $d(\chi) = |p| + 1 = d(a)$. □

Lemma 5.5.10. *Let ψ be an infinitary multistep and $\psi \xrightarrow[tgt_T]{\delta}\!\!\!\!\rightarrow \phi$. Then there exists a finite stepwise-or-nil proof term χ such that $\psi \approx_B \chi \cdot \phi$, $steps(\chi) = \mathsf{length}(\delta)$, and $d(\chi[i]) = d(\delta[i])$ for all $i < steps(\chi)$.*

Proof. Easy induction on $\mathsf{length}(\delta)$. If δ is an empty reduction sequence, then we conclude just by taking $\chi := src(\psi)$.

Assume $\delta = a; \delta'$, so that $\psi \xrightarrow[tgt_T]{a} \psi_0 \xrightarrow[tgt_T]{\delta'}\!\!\!\!\rightarrow \phi$. Then Lem. 5.5.9 implies that $\psi \approx_B \chi_0 \cdot \psi_0$ where χ_0 is a one-step verifying $d(\chi_0) = d(a)$, and IH on δ' yields $\psi_0 \approx_B \chi' \cdot \phi$ where χ' is a finite stepwise-or-nil proof term verifying $steps(\chi') = \mathsf{length}(\delta') = \mathsf{length}(\delta) - 1$, and $d(\chi'[i]) = d(\delta'[i]) = d(\delta[i+1])$ if $i < steps(\chi')$.

We take $\chi := \chi_0 \cdot \chi'$. It is straightforward to verify that χ satisfies the conditions about length and step depth. Moreover, $\psi_0 \approx_B \chi' \cdot \phi$ implies $\chi_0 \cdot \psi_0 \approx_B \chi_0 \cdot (\chi' \cdot \phi) \approx_B \chi \cdot \phi$, so that Trans yields $\psi \approx_B \chi \cdot \phi$ (recall $\psi \approx_B \chi_0 \cdot \psi_0$). Thus we conclude. □

The previous auxiliary results allow to prove the main result of this section, i.e. factorisation for infinitary multisteps.

Lemma 5.5.11. *Let ψ be a convergent infinitary multistep. Then there exist χ, ϕ such that $\psi \approx_B \chi \cdot \phi$, χ is a finite stepwise-or-nil proof term verifying $d(\chi[i]) = 0$ for all $i < steps(\chi)$, and ϕ is a convergent infinitary multistep verifying $mind(\phi) > 0$.*

Proof. We define $A := \{n \,/\, 0 < n < \omega$ and there is no collapsing sequence for ψ starting at ϵ and having length $n\}$. Dfn. 5.2.5 implies that ψ is tgt_T-weakly normalising. Then Lem. 5.5.7 implies that there is no infinite collapsing sequence for ψ starting at ϵ, so that Lem. 5.5.5 implies $A \neq \varnothing$. Let $n \in A$. Then Lem. 5.5.8 implies $\psi \xrightarrow[tgt_T]{\delta}\!\!\!\!\rightarrow \phi$, where $\mathsf{length}(\delta) < \omega$, $d(\delta[i]) = 0$ for all suitable i, and ϕ is an infinitary multistep (since it is the target of a tgt_T-reduction sequence) verifying $mind(\phi) > 0$ (since $\phi(\epsilon) \in \Sigma$). Moreover, ψ being convergent means that ψ is tgt_T-WN^∞, and tgt_T is a convergent iTRS, so that Lem. 5.1.50 implies that ϕ is also tgt_T-WN^∞, i.e. convergent. We conclude by applying Lem. 5.5.10 on $\psi \xrightarrow[tgt_T]{\delta}\!\!\!\!\rightarrow \phi$. □

5.5.2 Fixed prefix of contraction activity

This section introduces a technical tool, in which the extension of the factorisation result from infinitary multisteps to arbitrary proof terms is based on. This tool formalises a simple observation: the contraction activity denoted by a proof term can lie below some *fixed prefix*. Let us precise this idea, by means of an example using the rules $\mu : f(x) \to g(x)$, $\nu : g(x) \to k(x)$, and $\pi : a \to b$. The contraction activity corresponding to either of the equivalent proof terms $h(\mu(a) \cdot \nu(a), \pi)$ and $h(\mu(a), a) \cdot h(\nu(a), a) \cdot h(k(a), \pi)$ leaves the context $h(\Box, \Box)$ fixed, so we say that $h(\Box, \Box)$ is a fixed prefix for these proof terms. For proof terms involving root activity, the only possible fixed prefix is $\Box$. Note that the notion of "fixed prefix" is considered here in a strong sense, consistent with the strong convergence criterion we consider in this thesis, cfr. Section 5.1.4: a fixed prefix does not only coincide for the source and target of the involved contraction activity, it is furthermore *not affected* by that activity. In the sequel, we establish that fixed prefixes are invariant w.r.t. permutation equivalence.

Obtaining a *condensed-to-fixed-prefix-form* equivalent to a given proof term ψ allows to permute a step performed on $tgt(\psi)$, whose redex lies in the fixed prefix of ψ, by means of the (InOut) and (OutIn) equation schemas, as we do in the examples of permutation equivalence derivations described in Section 5.3.3. This observation is crucial in order to prove a general factorisation result, since it allows to obtain a proof term in which the (denotation of the) activity near to the root "shifts to the left as much as possible", i.e. lies in the lesser possible positions w.r.t. the order given by the sequence of dot occurrences in a proof term.

The following definitions and results characterise the common prefix of a proof term in a way allowing to manipulate it. The positions mentioned in the statements must be understood as being relative to the contraction activity denoted by a proof term, rather than as positions in the proof term. E.g. in the proof term $(h(\mu(a), a) \cdot h(\nu(a), a)) \cdot h(k(a), \pi)$, the three occurrences of h, which are at the positions 11, 12 and 2 in the proof term, correspond to the position ϵ in the denoted contraction activity; in turn, the occurrence of μ, at position 111 in the proof term, corresponds to the position 1 in the denoted activity. This assertion can be checked by observing the symbols corresponding, in the successive terms involved in the reduction sequence denoted by this proof term, namely $h(f(a), a) \to h(g(a), a) \to h(k(a), a) \to h(k(a), b)$, to the referred proof term occurrences.

We formalise the concept of (the activity denoted by) a proof term having a fixed prefix by defining a relation between proof terms and finite, prefix-closed sets of positions, which we call *respect*. Therefore, if ψ respects a set of positions P, then ψ has a fixed prefix corresponding to the positions in P.

Definition 5.5.12. *Let P be a set of positions, and $i \in \mathbb{N}$. Then we define the* projection of P on i *as* $P\,|_i := \{p \;/\; ip \in P\}$.

Definition 5.5.13. *Let t be a term, and P a finite and prefix-closed set of positions such that $P \subseteq \mathtt{Pos}(t)$. Then we define $t\,|^P$, the* prefix of t w.r.t. P, *as follows.*
If $P = \varnothing$, then $t\,|^P := \Box$.
If $P \neq \varnothing$ and $t \in \mathtt{Var}$, so that $P = \{\epsilon\}$, then $t\,|^P := t$.
If $P \neq \varnothing$ and $t = f(t_1, \ldots, t_m)$, so that $P = \{\epsilon\} \cup \bigcup_{1 \leqslant i \leqslant m}(i \cdot P\,|_i)$, then $t\,|^P := f(t_1\,|^{P\,|_1}, \ldots, t_m\,|^{P\,|_m})$.

Notice that $C = t\,|^P$ iff $t = C[t_1, \ldots, t_k]$ and $P = \{p \;/\; p \in \mathtt{Pos}(C) \wedge C(p) \neq \Box\}$, this can be verified by a simple induction on the cardinal of P.

Definition 5.5.14. *Let ψ be a proof term, and P a set of positions. We say that ψ* respects *P iff P is finite and prefix-closed, and any of the following applies:*

- *ψ is an infinitary multistep, $P \subseteq \texttt{Pos}(\psi)$ and $\psi(p) \in \Sigma$ for all $p \in P$.*
- *$\psi = \psi_1 \cdot \psi_2$ and both ψ_1 and ψ_2 respect P.*
- *$\psi = \cdot_{i<\omega} \psi_i$ and all ψ_i respect P.*
- *$\psi = f(\psi_1, \ldots, \psi_m)$, at least one of the ψ_i is not an infinitary multistep, and either $P = \varnothing$ or ψ_i respects $P\,|_i$ for all $i \leqslant m$.*
- *$\psi = \mu(\psi_1, \ldots, \psi_m)$, at least one of the ψ_i is not an infinitary multistep, and $P = \varnothing$.*

The relation just defined enjoys some simple properties.

Lemma 5.5.15. *Let ψ be a proof term and P such that ψ respects P. Then $P \subseteq \texttt{Pos}(src(\psi))$.*

Proof. An easy induction on ψ suffices; cfr. Prop. 5.2.16. □

Lemma 5.5.16. *Let ψ be a convergent proof term and P such that ψ respects P. Then $P \subseteq \texttt{Pos}(tgt(\psi))$.*

Proof. An easy induction on ψ suffices; cfr. Prop. 5.2.16. If $\psi = \cdot_{i<\omega} \psi_i$ and $P \subseteq \texttt{Pos}(tgt(\psi_i))$ for all $i < \omega$, given $p \in P$, we consider n such that $\texttt{dist}(tgt(\psi_i), tgt(\psi)) < 2^{-|p|}$ if $i > n$, so that $p \in \texttt{Pos}(tgt(\psi_{n+1}))$ implies $p \in \texttt{Pos}(tgt(\psi))$. □

Lemma 5.5.17. *Let $\psi = f(\psi_1, \ldots, \psi_m)$, and P a set of positions. Then ψ respects P iff either $P = \varnothing$ or ψ_i respects $P\,|_i$ for all $i \leqslant m$.*

Proof. If ψ is an infinitary multistep, then a straightforward analysis yields the desired result. If at least one of the ψ_i is not an infinitary multistep, then we conclude immediately. Any other case in Dfn. 5.5.14 contradicts the stated form of ψ. □

Lemma 5.5.18. *Let ψ be a proof term. Then ψ respects $\varnothing$.*

Proof. A straightforward induction on ψ, cfr. Prop. 5.2.16, suffices to conclude. □

The *respects* relation can be obtained from conditions on the target and the minimum activity depth of a proof term.

Lemma 5.5.19. *Let ψ be a convergent proof term and P a finite, prefix-closed set of positions, such that $mind(\psi) > n$, $|p| \leqslant n$ for all $p \in P$, and $P \subseteq \texttt{Pos}(tgt(\psi))$. Then ψ respects P.*

Proof. We proceed by induction on ψ, cfr. Prop. 5.2.16.

Assume that ψ is an infinitary multistep. If $P = \varnothing$ then Lem. 5.5.18 allows to conclude immediately. Otherwise, $\epsilon \in P$, implying $\psi = f(\psi_1, \ldots, \psi_m)$. We proceed by induction on n. If $n = 0$, then the only set of positions compatible with the lemma hypotheses is $P = \{\epsilon\}$, so that we conclude immediately. Assume $n = n' + 1$, and let i such that $1 \leqslant i \leqslant m$. It is straightforward to verify that $mind(\psi_i) > n'$, that $|p| \leqslant n'$ for all $p \in P\,|_i$, and also that $P\,|_i \subseteq \texttt{Pos}(tgt(\psi_i))$ (recall $tgt(\psi) = f(tgt(\psi_1), \ldots tgt(\psi_m))$). Therefore, we can apply IH on ψ_i, obtaining that ψ_i respects $P\,|_i$, so that $P\,|_i \subseteq \texttt{Pos}(\psi_i)$, and moreover for any $p \in P\,|_i$, $\psi(ip) = \psi_i(p) \in \Sigma$. Hence the desired result holds immediately.

Assume $\psi = \psi_1 \cdot \psi_2$. In this case, $mind(\psi_i) > n$ for $i = 1, 2$, and $P \subseteq \mathtt{Pos}(tgt(\psi)) = \mathtt{Pos}(tgt(\psi_2))$. Then IH applies to ψ_2 yielding that ψ_2 respects P. In turn, Lem. 5.5.15 implies $P \subseteq \mathtt{Pos}(src(\psi_2)) = \mathtt{Pos}(tgt(\psi_1))$. Then IH applies to ψ_1 as well, implying that ψ_1 respects P. Thus we conclude.

Assume $\psi = \cdot_{i<\omega} \psi_i$. Observe that $mind(\psi) > n$ implies $mind(\psi_i) > n$ for all $i < \omega$. Let k such that $\mathtt{dist}(tgt(\psi_i), tgt(\psi)) < 2^{-k}$ for all $i > k$. Let $j > k$. Then $P \subseteq \mathtt{Pos}(tgt(\psi))$ implies $P \subseteq \mathtt{Pos}(tgt(\psi_j))$. Then IH can be applied to ψ_j obtaining that ψ_j respects P. In turn, ψ_{k+1} respecting P implies that $P \subseteq \mathtt{Pos}(src(\psi_{k+1})) = \mathtt{Pos}(tgt(\psi_k))$. Therefore IH applies also to ψ_k, yielding that ψ_k respects P, and then Lem. 5.5.15 implies $P \subseteq \mathtt{Pos}(src(\psi_k)) = \mathtt{Pos}(tgt(\psi_{k-1}))$. Successive application of an analogous argument yields that ψ_i respects P for all $i \leqslant k$. Thus we conclude.

If $\psi = f(\psi_1, \ldots, \psi_m)$, then an argument analogous to that given for infinitary multisteps applies.

Finally, $\psi = \mu(\psi_1, \ldots, \psi_m)$ contradicts $mind(\psi) > n$ for any $n < \omega$. □

The *respects* relation is invariant w.r.t. base permutation equivalence.

Lemma 5.5.20. *Let ψ, ϕ be convergent proof terms and P a set of positions, such that $\psi \approx_B \phi$. Then ψ respects P iff ϕ respects P.*

Proof. We proceed by induction on α where $\psi \overset{\alpha}{\approx}_B \phi$, analysing the rule used in the last step of that judgement.

If the rule is Refl, then we conclude immediately.

If the rule is Eqn, then we analyse the equation used.

- (IdLeft) or (IdRight), i.e. $\psi = src(\phi) \cdot \phi$ or $\psi = \phi \cdot tgt(\phi)$. The $\Rightarrow$) direction is immediate. For the $\Leftarrow$) direction, observe that Lem. 5.5.15 and Lem. 5.5.16 imply $P \subseteq \mathtt{Pos}(src(\phi))$ and $P \subseteq \mathtt{Pos}(tgt(\phi))$ respectively. Then Dfn. 5.5.14 for infinitary multisteps implies immediately that both $src(\phi)$ and $tgt(\phi)$ respect P. Thus we conclude.

- (Assoc), i.e. $\psi = \chi_1 \cdot (\chi_2 \cdot \chi_3)$ and $\phi = (\chi_1 \cdot \chi_2) \cdot \chi_3$. In this case either ψ or ϕ respects P iff χ_1, χ_2 and χ_3 do. Thus we conclude.

- (Struct), i.e. $\psi = f(\chi_1, \ldots, \chi_m) \cdot f(\xi_1, \ldots, \xi_m)$ and $\phi = f(\chi_1 \cdot \xi_1, \ldots, \chi_m \cdot \xi_m)$. If $P = \varnothing$, then both ψ and ϕ respect P; cfr. Lem. 5.5.18. Otherwise
  ψ respects P
  iff both $f(\chi_1, \ldots, \chi_m)$ and $f(\xi_1, \ldots, \xi_m)$ do
  iff for all j such that $1 \leqslant j \leqslant m$, both χ_j and ξ_j respect $P|_j$
  iff for all j such that $1 \leqslant j \leqslant m$, $\chi_j \cdot \xi_j$ respects $P|_j$
  iff ϕ respects P.
  Thus we conclude.

- (InfStruct). This case admits an argument analogous to the one used for (Struct).

- (OutIn) and (InOut). In this case, it is immediate that either ψ or ϕ respects P iff $P = \varnothing$.

If the rule used in the last step of the judgement $\psi \overset{\alpha}{\approx}_B \phi$ is Symm, Trans, Fun, Comp or InfComp, then a straightforward inductive arguments suffices to obtain the desired result.

Finally, if the rule is Rule, then it is immediate to verify that either ψ or ϕ respect P iff $P = \varnothing$. □

Observe that proof terms whose minimum activity depth is greater than 0 are exactly those which respect $\{\epsilon\}$. Lem. 5.3.6 implies that this condition is stable by permutation equivalence. For such proof terms, we define their *condensed-to-fixed-prefix-symbol form*, which is a proof term denoting the same activity as the original proof term, and having a function symbol at the root. E.g. the condensed-to-fixed-prefix-symbol form of $f(\mu(a)) \cdot f(\nu(a))$ is $f(\mu(a) \cdot \nu(a))$. The condensed-to-fixed-prefix-symbol form of a condensed proof term is the same proof term, implying that this notion is idempotent.

Lemma 5.5.21. *Let ψ a convergent proof term which respects $\{\epsilon\}$. Then $src(\psi)(\epsilon) = tgt(\psi)(\epsilon)$.*

Proof. We proceed by induction on ψ, cfr. Prop. 5.2.16. If $\psi = f(\psi_1, \ldots, \psi_m)$ then the result holds immediately, while $\psi = \mu(\psi_1, \ldots, \psi_m)$ contradicts the lemma hypotheses.

If $\psi = \psi_1 \cdot \psi_2$ and the result holds for both components, then lemma hypotheses imply that both ψ_1 and ψ_2 respect $\{\epsilon\}$, so that $src(\psi_j)(\epsilon) = tgt(\psi_j)(\epsilon)$ for $j = 1, 2$. Observe $src(\psi) = src(\psi_1)$, $tgt(\psi) = tgt(\psi_2)$, and moreover $tgt(\psi_1) = src(\psi_2)$ (by the coherence condition on the definition of ψ). Thus we conclude immediately.

Assume $\psi = \cdot_{i<\omega} \psi_i$ and the result holds for each ψ_i. For any $i < \omega$, lemma hypotheses imply that ψ_i respects $\{\epsilon\}$, and therefore $src(\psi_i)(\epsilon) = tgt(\psi_i)(\epsilon)$. Given $tgt(\psi_i) = src(\psi_{i+1})$ for all $i < \omega$, an easy inductive argument yields $src(\psi)(\epsilon) = src(\psi_0)(\epsilon) = tgt(\psi_i)(\epsilon)$ for any $i < \omega$. Let n such that $\mathtt{dist}(tgt(\psi_k), tgt(\psi)) < 1$ if $k > n$; recall $tgt(\psi) = \lim_{i\to\omega}(tgt(\psi_i))$. Then $tgt(\psi)(\epsilon) = tgt(\psi_{n+1})(\epsilon) = src(\psi)(\epsilon)$. Thus we conclude. □

Definition 5.5.22. *Let ψ be a proof term which respects $\{\epsilon\}$. We define $\mathsf{cfps}(\psi)$, i.e. the* condensed to fixed prefix symbol form *of ψ, as follows.*

- *if* $\psi = f(\psi_1, \ldots, \psi_n)$ *then* $\mathsf{cfps}(\psi) := \psi$.
- *if* $\psi = \psi_1 \cdot \psi_2$ *then* $\mathsf{cfps}(\psi) := f(\psi_{11} \cdot \psi_{21}, \ldots, \psi_{1m} \cdot \psi_{2m})$ *where* $\mathsf{cfps}(\psi_i) = f(\psi_{i1}, \ldots, \psi_{im})$ *for* $i = 1, 2$
- *if* $\psi = \cdot_{i<\omega} \psi_i$ *then* $\mathsf{cfps}(\psi) := f(\cdot_{i<\omega} \psi_{i1}, \ldots, \cdot_{i<\omega} \psi_{im})$ *where* $\mathsf{cfps}(\psi_i) = f(\psi_{i1}, \ldots, \psi_{im})$ *for all* $i < \omega$.
- $\psi = \mu(\psi_1, \ldots, \psi_m)$ *contradicts* ψ *respecting* $\{\epsilon\}$.

Lem. 5.5.21 implies the soundness of the clauses corresponding to both binary and infinite concatenation.

Condensed-to-fixed-prefix-symbol forms enjoy some properties related with base permutation equivalence and the *respects* relation. In turn, these properties allow a simple proof of the extension of Lem. 5.5.21 to arbitrary finite and prefix-closed sets of positions.

Lemma 5.5.23. *Let ψ be a proof term which respects $\{\epsilon\}$. Then $\psi \approx_B \mathsf{cfps}(\psi)$.*

Proof. Easy induction on ψ. For the infinitary composition case, resort to the InfComp rule and the (InfStruct) equation, cfr. Dfn. 5.3.3. □

Lemma 5.5.24. *Let ψ, ϕ be proof terms such that $\psi \approx_B \phi$ and ψ, ϕ respect $\{\epsilon\}$. Let $\mathsf{cfps}(\psi) = f(\psi_1, \ldots, \psi_m)$ and $\mathsf{cfps}(\phi) = f'(\phi_1, \ldots, \phi_{m'})$. Then $f = f' = src(\psi)(\epsilon)$, so that $m = m'$, and $\psi_i \approx_B \phi_i$ for each i between 1 and m.*

Proof. Lem. 5.5.23 and the hypotheses imply $\psi \approx_B \mathsf{cfps}(\psi) \approx_B \mathsf{cfps}(\phi)$, then Lem. 5.3.6 yields $f = f' = src(\psi)(\epsilon)$, and therefore $m = m'$. We prove $\psi_i \approx_B \phi_i$ for all i by induction on α where $\psi \overset{\alpha}{\approx}_B \phi$, analysing the rule used in the last step of that judgement.

- Refl: we conclude immediately.
- Eqn: we analyse each of the equations.
  - (IdLeft): let $src(\phi) = f(t_1, \ldots, t_m)$ where $t_i = src(\phi_i)$ for all i; cfr. Lem. 5.5.23 and Lem. 5.3.6. Then $\psi = f(t_1, \ldots, t_m) \cdot \phi$, so that $\mathsf{cfps}(\psi) = f(t_1 \cdot \phi_1, \ldots, t_m \cdot \phi_m)$. Thus we conclude.
  - (IdRight): an analogous argument applies.
  - (Assoc): in this case $\psi = \xi \cdot (\gamma \cdot \chi)$ and $\phi = (\xi \cdot \gamma) \cdot \chi$. Let $\mathsf{cfps}(\xi) = f(\xi_1, \ldots, \xi_m)$, $\mathsf{cfps}(\gamma) = f(\gamma_1, \ldots, \gamma_m)$ and $\mathsf{cfps}(\chi) = f(\chi_1, \ldots, \chi_m)$; cfr. Lem. 5.5.23 (implying $f = src(\psi)(\epsilon) = src(\xi)(\epsilon) = src(\mathsf{cfps}(\xi))(\epsilon)$) and Lem. 5.5.21. Then for any $i \leqslant m$, $\psi_i = \xi_i \cdot (\gamma_i \cdot \xi_i)$ and $\phi_i = (\xi_i \cdot \gamma_i) \cdot \chi_i$. Thus we conclude immediately.
  - (Struct) and (InfStruct): in either of these cases Dfn. 5.5.22 allows to conclude immediately.
  - (OutIn) and (InOut): either of these cases contradict ψ, ϕ to respect $\{\epsilon\}$.
- Symm or Trans: a simple inductive argument applies.
- Fun: the hypotheses of the Fun rule are enough to conclude immediately.
- Rule: this case would imply that neither ψ nor ϕ respect $\{\epsilon\}$, thus contradicting lemma hypotheses.
- Comp: in this case, $\psi = \chi \cdot \xi$, $\phi = \gamma \cdot \delta$, $\chi \overset{\alpha_1}{\approx}_B \gamma$, $\xi \overset{\alpha_2}{\approx}_B \delta$, $\alpha_1 < \alpha$ and $\alpha_2 < \alpha$. Let $\mathsf{cfps}(\chi) = f(\chi_1, \ldots, \chi_m)$, $\mathsf{cfps}(\xi) = f(\xi_1, \ldots, \xi_m)$, $\mathsf{cfps}(\gamma) = f(\gamma_1, \ldots, \gamma_m)$ and $\mathsf{cfps}(\delta) = f(\delta_1, \ldots, \delta_m)$. Let i such that $1 \leqslant i \leqslant m$. Observe $\psi_i = \chi_i \cdot \xi_i$ and $\phi_i = \gamma_i \cdot \delta_i$. On the other hand, IH implies $\chi_i \approx_B \gamma_i$ and $\xi_i \approx_B \delta_i$. Thus we conclude.
- InfComp: an analogous argument applies. In this case, $\psi = \cdot_{i<\omega}\, \psi_i$, $\phi = \cdot_{i<\omega}\, \phi_i$, and for any $i < \omega$, $\psi_i \overset{\alpha_i}{\approx}_B \phi_i$ where $\alpha_i < \alpha$. Let $\mathsf{cfps}(\psi_i) = f(\psi_i^1, \ldots, \psi_i^m)$ and $\mathsf{cfps}(\phi_i) = f(\phi_i^1, \ldots, \phi_i^m)$. Let j such that $1 \leqslant j \leqslant m$. Then $\psi_j = \cdot_{i<\omega}\, \psi_i^j$ and $\phi_j = \cdot_{i<\omega}\, \phi_i^j$. IH on each $\psi_i \overset{\alpha_i}{\approx}_B \phi_i$ yields $\psi_i^j \approx_B \phi_i^j$. Thus we conclude.

□

Lemma 5.5.25. *Let ψ be a proof term such that ψ respects $\{\epsilon\}$. Then $\mathsf{cfps}(\psi)(\epsilon) = src(\psi)(\epsilon) = tgt(\psi)(\epsilon)$.*

Proof. Immediate consequence of Lem. 5.5.24 and Lem. 5.5.21. □

Lemma 5.5.26. *Let ψ be a proof term and P a set of positions such that $P \neq \varnothing$ and ψ respects P. Then ψ_i respects $P|_i$ for all $i \leqslant m$, where $\mathsf{cfps}(\psi) = f(\psi_1, \ldots, \psi_m)$.*

Proof. Lem. 5.5.23 implies $\psi \approx_B \mathsf{cfps}(\psi)$, then Lem. 5.5.20 implies $\mathsf{cfps}(\psi)$ respects P. Therefore Lem. 5.5.17 allows to conclude. □

Lemma 5.5.27. *Let ψ be a convergent proof term and P a set of positions such that ψ respects P. Then $tgt(\psi)\,|^P = src(\psi)\,|^P$.*

Proof. We proceed by induction on the cardinal of P. If $P = \varnothing$, then $tgt(\psi)\,|^P = src(\psi)\,|^P = \Box$. Otherwise, $P = \{\epsilon\} \cup (\bigcup_{1 \leqslant i \leqslant m} i \cdot P\,|_i)$ where $\mathsf{cfps}(\psi) = f(\psi_1, \ldots, \psi_m)$. In this case, Lem. 5.5.23 and Lem. 5.3.6 imply $src(\psi) = f(src(\psi_1), \ldots, src(\psi_m))$ and $tgt(\psi) = f(tgt(\psi_1), \ldots, tgt(\psi_m))$, so that $src(\psi)\,|^P = f(src(\psi_1)\,|^{P\,|_1}, \ldots, src(\psi_m)\,|^{P\,|_m})$, and $tgt(\psi)\,|^P = f(tgt(\psi_1)\,|^{P\,|_1}, \ldots, tgt(\psi_m)\,|^{P\,|_m})$. On the other hand, Lem. 5.5.26 implies that ψ_i respects $P\,|_i$ for all i, so that IH can be applied to obtain $src(\psi_i)\,|^{P\,|_i} = tgt(\psi_i)\,|^{P\,|_i}$. Thus we conclude. $\Box$

Assume that some proof term, say ψ, respects not only the root, but a finite, prefix-closed set of positions P. Then we can define the *condensed-to-fixed-prefix-**context** form* of ψ w.r.t. P, analogously as we have just done with the condensed-to-fixed-prefix-symbol form. The activity denoted by a condensed-to-fixed-prefix-context form w.r.t. the set of positions P will lie inside a fixed context, i.e. a context in $Ter(\Sigma)$, whose set of (non-hole) positions is exactly P. E.g., the proof term $h(f(g(\mu(a))), \mu(b)) \cdot h(f(g(g(\pi))), \nu(b))$ respects $P := \{\epsilon, 1, 11\}$. The corresponding condensed-to-fixed-prefix-context is $h(f(g(\mu(a) \cdot g(\pi))), \mu(b) \cdot \nu(b))$. Observe that the activity of the latter term lies inside the holes of the context $h(f(g(\Box)), \Box)$, whose set of non-hole positions is exactly P.

The condensed-to-fixed-prefix-context form of ψ w.r.t. P can be defined in two different ways: either by induction on ψ analogously as the definition of cfps, or by induction on P. The following definition uses the latter option for a pragmatic reason: it leads to simpler proofs of the properties to be stated about these forms.

Definition 5.5.28. *Let ψ be a proof term and P a prefix-closed set of positions, such that ψ respects P. We define $\mathsf{cfpc}(\psi, P)$, the* condensed to fixed prefix context form *of ψ w.r.t. P, as follows.*
If $P = \varnothing$, then $\mathsf{cfpc}(\psi, P) := \psi$.
Otherwise, $P = \{\epsilon\} \cup (\bigcup_{1 \leqslant i \leqslant m} i \cdot P\,|_i)$, where $src(\psi)(\epsilon) = f/m$. In this case $\mathsf{cfpc}(\psi, P) := f(\mathsf{cfpc}(\psi_1, P\,|_1), \ldots \mathsf{cfpc}(\psi_m, P\,|_m))$, where $\mathsf{cfps}(\psi) = f(\psi_1, \ldots, \psi_m)$.

Lemma 5.5.29. *Let ψ, P such that ψ respects P. Then $\psi \approx_B \mathsf{cfpc}(\psi, P)$.*

Proof. We proceed by induction on the cardinal of P. If $P = \varnothing$ then we conclude immediately. Otherwise, $P = \{\epsilon\} \cup (\bigcup_{1 \leqslant i \leqslant m} i \cdot P\,|_i)$ where $\mathsf{cfps}(\psi) = f(\psi_1, \ldots, \psi_m)$, and $\mathsf{cfpc}(\psi, P) = f(\mathsf{cfpc}(\psi_1, P\,|_1), \ldots, \mathsf{cfpc}(\psi_m, P\,|_m))$. Lem. 5.5.26 implies that ψ_i respects $P\,|_i$ for all $i \leqslant m$. Therefore IH can be applied on each $P\,|_i$ to obtain $\psi_i \approx_B \mathsf{cfpc}(\psi_i, P\,|_i)$, so that Fun rule yields $\mathsf{cfps}(\psi) \approx_B \mathsf{cfpc}(\psi, P)$. On the other hand, Lem. 5.5.23 implies $\psi \approx \mathsf{cfps}(\psi)$. Thus we conclude by Trans. $\Box$

Lemma 5.5.30. *Let ψ, ϕ, P such that ψ and ϕ are convergent, $\psi \approx_B \phi$ and ψ, ϕ respect P. Then $\mathsf{cfpc}(\psi, P) = C[\psi_1, \ldots, \psi_k]$, $\mathsf{cfpc}(\phi, P) = C[\phi_1, \ldots, \phi_k]$ and $\psi_i \approx_B \phi_i$ for all i, where $C = src(\psi)\,|^P$.*

Proof. We proceed by induction on the cardinal of P. If $P = \varnothing$ then we conclude immediately. Otherwise $P = \{\epsilon\} \cup (\bigcup_{1 \leqslant i \leqslant m} i \cdot P\,|_i)$, $\mathsf{cfpc}(\psi, P) = f(\mathsf{cfpc}(\psi'_1, P\,|_1), \ldots, \mathsf{cfpc}(\psi'_m, P\,|_m))$, and $\mathsf{cfpc}(\phi, P) = f(\mathsf{cfpc}(\phi'_1, P\,|_1), \ldots, \mathsf{cfpc}(\phi'_m, P\,|_m))$, where $\mathsf{cfps}(\psi) = f(\psi'_1, \ldots, \psi'_m)$ and $\mathsf{cfps}(\phi) = f(\phi'_1, \ldots, \phi'_m)$. Lem. 5.5.23 and Lem. 5.3.6 imply that

$src(\psi) = f(src(\psi'_1), \ldots, src(\psi'_m))$ and analogously for ϕ, so that particularly the root symbols of $\mathsf{cfps}(\psi)$ and $\mathsf{cfps}(\phi)$ coincide since $\psi \approx_B \phi$.

Let j such that $1 \leqslant j \leqslant m$. Lem. 5.5.24 implies that $\psi'_j \approx_B \phi'_j$, and Lem. 5.5.26 implies that both ψ'_j and ϕ'_j respect $P\,|_j$. Then we can apply IH on $P\,|_j$ obtaining that $\mathsf{cfpc}(\psi'_j, P\,|_j) = C_j[\psi^j_1, \ldots, \psi^j_{q_j}]$, $\mathsf{cfpc}(\phi'_j, P\,|_j) = C_j[\phi^j_1, \ldots, \phi^j_{q_j}]$ and $\psi^j_i \approx_B \phi^j_i$ for all i, where $src(\psi'_j)\,|^{P\,|_j} = C_j$.

We define $C := f(C_1, \ldots, C_m)$. It is straightforward to verify that $src(\psi)\,|^P = C$. Moreover, $\mathsf{cfpc}(\psi, P) = C[\psi_1, \ldots, \psi_k]$ and $\mathsf{cfpc}(\phi, P) = C[\phi_1 \ldots, \phi_k]$, where $k = \sum_{1 \leqslant i \leqslant m} q_i$, and for any $i \leqslant k$, $\psi_i = \psi^j_l$ and $\phi_i = \phi^j_l$ for some $j \leqslant m$ and $l \leqslant q_j$, implying $\psi_i \approx_B \phi_i$. Thus we conclude. □

Lemma 5.5.31. *Let ψ, P such that ψ respects P. Then $\mathsf{cfpc}(\psi, P)\,|^P = src(\psi)\,|^P = tgt(\psi)\,|^P$.*

Proof. Straightforward corollary of Lem. 5.5.30 and Lem. 5.5.27. □

5.5.3 General factorisation result

In this section we will extend the factorisation result obtained for infinitary multisteps in Sec. 5.5.1, to the set of all proof terms. As we have already mentioned, the condensed-to-proof-term forms introduced in Sec. 5.5.2 lead to the proof of the main remaining auxiliary result, namely, the ability of obtain proof terms in which activity at lower depths is in low positions w.r.t. the sequentialisation order given by dot occurrences.

Lemma 5.5.32. *Let ψ be a one-step. Then there exist two numbers $n, n' < \omega$ such that, for any convergent proof term ξ verifying $tgt(\xi) = src(\psi)$ and $mind(\xi) \geqslant n + n'$, a one-step ψ' and a convergent proof term ξ' can be found, which verify all the following: $\xi \cdot \psi \approx_B \psi' \cdot \xi'$, $d(\psi') = d(\psi)$, and $mind(\xi') \geqslant mind(\xi) - n'$.*

Proof. We take $n := d(\psi)$ and $n' = \mathtt{Pd}(\mu) + 1$ where $\mu := \psi(\mathtt{rpos}(\psi))$. We consider a convergent proof term ξ verifying $mind(\xi) \geqslant n + n'$ and $tgt(\xi) = src(\psi)$.

Let $P_0 := \{p \;/\; p \in src(\psi) \wedge |p| < d(\psi)\}$, $P := P_0 \cup (\mathtt{rpos}(\psi) \cdot \mathtt{Ppos}(\mu))$, and $k := max\{|p| \;/\; p \in P\}$. Observe that $p \in P$ implies $|p| \leqslant d(\psi) + \mathtt{Pd}(\mu)$, so that $k \leqslant d(\psi) + \mathtt{Pd}(\mu) < mind(\xi)$. Moreover, it is straightforward to verify that $P \subseteq \mathtt{Pos}(src(\psi)) = \mathtt{Pos}(tgt(\xi))$. Therefore Lem. 5.5.19 applies w.r.t. ξ, P and k, implying that ξ respects P. Then $\xi_F := \mathsf{cfpc}(\xi, P)$ can be defined. In turn, Lem. 5.5.29 implies that $\xi \approx_B \xi_F$, so that $\xi \cdot \psi \approx_B \xi_F \cdot \psi$, and Lem. 5.5.31 implies $\xi_F\,|^P = tgt(\xi)\,|^P = src(\psi)\,|^P$.

Let $C := src(\psi)\,|^{P_0}$. An easy induction on $d(\psi)$ yields that $\psi\,|^{P_0} = C$, so that the comment following Dfn. 5.5.13 implies $\psi = C[t_1, \ldots, t_{j-1}, \mu(u_1, \ldots, u_m), t_{j+1}, \ldots, t_k]$ and $\{p \;/\; p \in \mathtt{Pos}(C) \wedge C(p) \neq \Box\} = P_0$. Observe that $|\mathtt{Bpos}(C, i)| = d(\psi)$ for all i, and that particularly $\mathtt{Bpos}(C, j) = \mathtt{rpos}(\psi)$ for some j. In turn, the given form of ψ implies that $src(\psi) = tgt(\xi) = C[t_1, \ldots, t_{j-1}, l[u_1, \ldots, u_m], t_{j+1}, \ldots, t_k]$ where $\mu : l \to h$. Observe that the set of non-hole positions of the context $C[\Box, \ldots, \Box, l[\Box, \ldots, \Box], \Box, \ldots, \Box]$ is exactly P, implying that $C = tgt(\xi)\,|^P = \xi_F\,|^P$, and therefore $\xi_F = C[\xi_1, \ldots, \xi_{j-1}, l[\phi_1, \ldots, \phi_m], \xi_{j+1}, \ldots, \xi_k]$; cfr. the comment following Dfn. 5.5.13. Notice that ξ_F is convergent, implying that all the ξ_i and also the ϕ_i are; cfr. Lem. 5.3.7 and Lem. 5.2.21. Moreover, $t_i = tgt(\xi_i)$ for any suitable i, and also $u_i = tgt(\phi_i)$ for all suitable i. Hence

$$\begin{array}{ll}\xi_F \cdot \psi & \\ \approx_B & C[\xi_1 \cdot t_1, \ldots, \xi_{j-1} \cdot t_{j-1}, l[\phi_1 \ldots \phi_m] \cdot \mu(u_1, \ldots, u_m), \xi_{j+1} \cdot t_{j+1}, \ldots, \xi_k \cdot t_k] \\ \approx_B & C[\xi_1, \ldots, \xi_{j-1}, \mu(\phi_1, \ldots, \phi_m), \xi_{j+1}, \ldots, \xi_k] \\ \approx_B & C[s_1 \cdot \xi_1, \ldots, s_{j-1} \cdot \xi_{j-1}, \mu(w_1, \ldots, w_m) \cdot h[\phi_1, \ldots, \phi_m], s_{j+1} \cdot \xi_{j+1}, \ldots, s_k \cdot \xi_k] \\ \approx_B & C[s_1, \ldots, s_{j-1}, \mu(w_1, \ldots, w_m), s_{j+1}, \ldots, s_k] \cdot \\ & C[\xi_1, \ldots, \xi_{j-1}, h[\phi_1, \ldots, \phi_m], \xi_{j+1}, \ldots, \xi_k]\end{array}$$

where $s_i := src(\xi_i)$ and $w_i := src(\phi_i)$, in both cases for all suitable i. To justify the equivalences; cfr. Lem. 5.3.9; (IdRight), (InOut) and Lem. 5.3.8; (IdLeft), (OutIn) and Lem. 5.3.8 again; and finally Lem. 5.3.9 again; respectively.

We take $\psi' := C[s_1, \ldots, s_{j-1}, \mu(w_1, \ldots, w_m), s_{j+1}, \ldots, s_k]$ and $\xi' := C[\xi_1, \ldots, \xi_{j-1}, h[\phi_1, \ldots, \phi_m], \xi_{j+1}, \ldots, \xi_k]$. Observe that convergence of all ξ_i and ϕ_i imply convergence of ξ', cfr. Lem. 5.2.21.

In order to conclude, we must verify that $mind(\xi') \geqslant mind(\xi) - n' = mind(\xi_F) - (\mathtt{Pd}(\mu) + 1)$; cfr. Lem. 5.3.6. Let a such that $mind(\xi_a) \leqslant mind(\xi_i)$ for all i such that $1 \leqslant i \leqslant k$ and $i \neq j$, b such that $mind(\phi_b) + |\mathtt{Bpos}(l, b)| \leqslant mind(\phi_i) + |\mathtt{Bpos}(l, i)|$ for all i such that $1 \leqslant i \leqslant m$, and c, k such that $mind(\phi_c) + |\mathtt{Bpos}(h, k)| \leqslant mind(\phi_i) + |\mathtt{Bpos}(h, j)|$ if $1 \leqslant i \leqslant m$ and $h(\mathtt{Bpos}(h, j)) = x_i$. In these definitions, l and h are considered as contexts as when we write e.g. $l[\phi_1, \ldots, \phi_m]$. Lem. 5.2.18 implies $mind(\xi_F) = d(\psi) + min(mind(\xi_a), mind(\phi_b) + |\mathtt{Bpos}(l, b)|)$ and $mind(\xi') = d(\psi) + min(mind(\xi_a), mind(\phi_c) + |\mathtt{Bpos}(h, k)|)$. Observe that $|\mathtt{Bpos}(l, i)| \leqslant \mathtt{Pd}(\mu) + 1$ for all i. We show $mind(\xi_F) - (\mathtt{Pd}(\mu) + 1) \leqslant mind(\xi')$.

If $mind(\xi_a) \leqslant mind(\phi_c) + |\mathtt{Bpos}(h, k)|$, then $mind(\xi_F) \leqslant d(\psi) + mind(\xi_a) = mind(\xi')$ in either case w.r.t. the characterisation of $mind(\xi_F)$. Otherwise, i.e. if $mind(\phi_c) + |\mathtt{Bpos}(h, k)| < mind(\xi_a)$, observe that $mind(\xi_F) \leqslant d(\psi) + mind(\phi_b) + |\mathtt{Bpos}(l, b)|$ holds in any case. Therefore

$$\begin{array}{lll} mind(\xi_F) & \leqslant & d(\psi) + mind(\phi_b) + |\mathtt{Bpos}(l, b)| \\ & \leqslant & d(\psi) + mind(\phi_c) + |\mathtt{Bpos}(l, c)| \\ & \leqslant & d(\psi) + mind(\phi_c) + (\mathtt{Pd}(\mu) + 1) \end{array}$$

Therefore $mind(\xi_F) - (\mathtt{Pd}(\mu) + 1) \leqslant d(\psi) + mind(\phi_c) \leqslant d(\psi) + mind(\phi_c) + |\mathtt{Bpos}(h, k)| = mind(\xi')$. □

Lemma 5.5.33. *Let ψ be a finite stepwise-or-nil proof term. Then there exist two numbers $n, n' < \omega$ such that, for any convergent proof term ξ verifying $tgt(\xi) = src(\psi)$ and $mind(\xi) \geqslant n + n'$, a finite stepwise-or-nil proof term ψ' and a convergent proof term ξ' can be found, which verify all the following: $\xi \cdot \psi \approx_B \psi' \cdot \xi'$, $steps(\psi') = steps(\psi)$, $d(\psi'[i]) = d(\psi[i])$ for all i, and $mind(\xi') \geqslant mind(\xi) - n' \geqslant n$.*

Proof. We proceed by induction on $steps(\psi)$. If $steps(\psi) = 0$, i.e. $\psi \in Ter^{\infty}(\Sigma)$, then $src(\psi) = \psi$. Therefore we can take $n = n' = 0$, since for any ξ verifying $tgt(\xi) = \psi$, it is straightforward to obtain $\xi \cdot \psi \approx_B src(\xi) \cdot \xi$, and to verify the required properties for $\psi' := src(\xi)$ and $\xi' := \xi$.

Assume $steps(\psi) = n + 1$, i.e. $\psi = \chi \cdot \phi$, where χ is a one-step and ϕ is a stepwise-or-nil proof term verifying $steps(\phi) = n$. In this case, IH can be applied on ϕ; let m and m' be the corresponding numbers. Moreover, Lem. 5.5.32 applies to χ; let p and p' be the numbers whose existence is stated by that lemma. Let $n := max(m, p)$ and $n' := m' + p'$. Let ξ a convergent proof term verifying $mind(\xi) \geqslant n + n' = n + m' + p' \geqslant p + p'$, and $tgt(\xi) = src(\psi) = src(\chi)$. Then the conclusion of Lem. 5.5.32 implies that $\xi \cdot \psi = \xi \cdot \chi \cdot \phi \approx_B \chi' \cdot \xi'' \cdot \phi$, where χ' is a one-step verifying $d(\chi') = d(\chi)$ and ξ'' is a convergent proof term such that $mind(\xi'') \geqslant mind(\xi) - p' \geqslant n + m' \geqslant m + m'$.

In turn, the conclusion of the IH implies that $\chi' \cdot \xi'' \cdot \phi \approx_B \chi' \cdot \phi' \cdot \xi'$, where ϕ' is a stepwise-or-nil proof term verifying $steps(\phi') = steps(\phi)$ and $d(\phi'[i]) = d(\phi[i])$ for all i, and ξ' is a convergent proof term such that $mind(\xi') \geqslant mind(\xi'') - m' \geqslant n$. We take $\psi' := \chi' \cdot \phi'$, and we conclude by observing that Trans implies $\xi \cdot \psi \approx_B \psi' \cdot \xi'$. □

The given auxiliary results allow to prove the statement being the aim of this Section.

Proposition 5.5.34. *Let ψ be a convergent proof term and $n < \omega$. Then there exist χ and ϕ such that $\psi \approx_B \chi \cdot \phi$, χ is a finite stepwise-or-nil proof term, ϕ is convergent and $mind(\phi) > n$.*

Proof. We proceed by induction on α where $\psi \in \mathbf{PT}_\alpha$, analysing the cases in the formation of ψ w.r.t. Dfn. 5.2.8.

- Assume that ψ is an infinitary multistep. In this case we proceed by induction on n. If $n = 0$ then Lem. 5.5.11 suffices to conclude.

  Assume $n = n' + 1$. Lem. 5.5.11 implies $\psi \approx_B \chi_0 \cdot \phi'$ where χ_0 is a finite stepwise-or-nil proof term, ϕ' is a convergent infinitary multistep and $mind(\phi') > 0$, so that $\phi' = f(\phi'_1, \ldots, \phi'_m)$. Observe that ϕ' convergent implies ϕ'_i convergent for all i, cfr. Lem. 5.2.19. Then IH can be applied on all ϕ'_i w.r.t. n', yielding $\phi' \approx_B f(\chi_1 \cdot \phi_1, \ldots, \chi_m \cdot \phi_m)$ where for all i, χ_i is a finite stepwise-or-nil proof term, ϕ_i is convergent and $mind(\phi_i) > n'$. Hence $\psi \approx_B \chi_0 \cdot f(\chi_1, \ldots, \chi_m) \cdot f(\phi_1, \ldots, \phi_m)$.

  Assume that $m = 3$; observe that $f(\chi_1, \chi_2, \chi_3) \approx_B f(\chi_1 \cdot t_1, s_2 \cdot \chi_2, s_3 \cdot \chi_3) \approx_B f(\chi_1, s_2, s_3) \cdot f(t_1, \chi_2, \chi_3) \approx_B f(\chi_1, s_2, s_3) \cdot f(t_1 \cdot t_1, \chi_2 \cdot t_2, s_3 \cdot \chi_3) \approx_B f(\chi_1, s_2, s_3) \cdot f(t_1, \chi_2, s_3) \cdot (t_1, t_2, \chi_3)$. An analogous reasoning for any m yields that $f(\chi_1, \chi_2, \ldots, \chi_m) \approx_B f(\chi_1, src(\chi_2), \ldots, src(\chi_m)) \cdot f(tgt(\chi_1), \chi_2, \ldots, src(\chi_m)) \cdot f(tgt(\chi_1), tgt(\chi_2), \ldots, \chi_m)$. In turn, it is straightforward to obtain a stepwise proof term $\chi'_k \approx_B f(tgt(\chi_1), \ldots, \chi_k, \ldots, src(\chi_m))$, so that $\chi' := \chi'_0 \cdot \ldots \cdot \chi'_m$ is a stepwise proof term verifying $\chi' \approx_B f(\chi_1, \chi_2, \ldots, \chi_m)$. Thus we conclude by taking $\chi := \chi_0 \cdot \chi'$ and $\phi := f(\phi_1, \ldots, \phi_m)$.

- Assume $\psi = \psi_1 \cdot \psi_2$ and ψ is not an infinite composition. In this case we can apply IH on ψ_2, obtaining $\psi_2 \approx_B \chi_2 \cdot \phi_2$ where χ_2 is a finite stepwise-or-nil proof term, ϕ_2 is convergent and $mind(\phi_2) > n$. Lem. 5.5.33 applies to χ_2, implying the existence of two numbers, say m_0 and m', which enjoy some properties. Let $m := max(n, m_0)$. Applying IH on ψ_1 w.r.t. $m + m'$, we obtain $\psi_1 \approx_B \chi_1 \cdot \phi_1$, where χ_1 is a finite stepwise-or-nil proof term, ϕ_1 is convergent and $mind(\phi_1) > m + m' \geqslant m_0 + m'$. Observe $\psi \approx_B \chi_1 \cdot \phi_1 \cdot \chi_2 \cdot \phi_2$, so that $tgt(\phi_1) = src(\chi_2)$.

  Therefore, the conclusion of Lem. 5.5.33 implies $\phi_1 \cdot \chi_2 \approx_B \chi'_2 \cdot \phi'_1$, so that $\psi \approx_B \chi_1 \cdot \chi'_2 \cdot \phi'_1 \cdot \phi_2$, where χ'_2 is a finite stepwise-or-nil proof term (since $steps(\chi'_2) = steps(\chi_2)$), ϕ'_1 is convergent and $mind(\phi'_1) \geqslant mind(\phi_1) - m' > m \geqslant n$. Thus we conclude by taking $\chi := \chi_1 \cdot \chi'_2$ and $\phi := \phi'_1 \cdot \phi_2$.

- Assume $\psi = \cdot_{i<\omega}\, \psi_i$. Let k such that $mind(\psi_i) > n$ if $i > k$; convergence of ψ entails the existence of such k. Then $\psi \approx_B \psi_0 \cdot \ldots \cdot \psi_k \cdot (\cdot_{i<\omega}\, \psi_{k+1+i})$, and $mind(\cdot_{i<\omega}\, \psi_{k+1+i}) > n$; notice that convergence of ψ implies convergence of $\cdot_{i<\omega}\, \psi_{k+1+i}$. Observe that $\psi_0 \cdot \ldots \cdot \psi_k \in \mathbf{PT}_{\alpha'}$ where $\alpha' < \alpha$. This observation allows to use IH to obtain $\psi_0 \cdot \ldots \cdot \psi_k \approx_B \chi \cdot \phi'$ where χ is a finite stepwise-or-nil proof term, ϕ' is convergent and $mind(\phi') > n$. Then we conclude by taking $\phi := \phi' \cdot (\cdot_{i<\omega}\, \psi_{k+1+i})$.

- Assume $\psi = f(\psi_1, \ldots, \psi_m)$ and ψ is not an infinitary multistep. In this case, we can apply IH on each ψ_i obtaining $\psi_i \approx_B \chi_i \cdot \phi_i$, where χ_i is a finite stepwise-or-nil proof term, ϕ_i is convergent, and $mind(\phi_i) > n$. Then $\psi \approx_B f(\chi_1, \ldots, \chi_m) \cdot f(\phi_1, \ldots, \phi_m)$. Hence, an argument about $f(\chi_1, \ldots, \chi_m)$ analogous to that used in the infinitary multistep case allows to conclude.

- Assume $\psi = \mu(\psi_1, \ldots, \psi_m)$ and ψ is not an infinitary multistep. Let us define $\mu : l[x_1, \ldots, x_m] \to h$.

  Assume $h = f(h_1, \ldots, h_k)$. In this case we have $\psi \approx_B \mu(src(\psi_1), \ldots, src(\psi_m)) \cdot f(h_1[\psi_1, \ldots, \psi_m], \ldots, h_k[\psi_1, \ldots, \psi_m])$. Applying IH on each ψ_i yields $\psi_i \approx_B \chi_i \cdot \phi_i$, where χ_i is a finite stepwise-or-nil proof term, ϕ_i is convergent, and $mind(\phi_i) > n$.
  Therefore $\psi \approx_B \mu(src(\psi_1), \ldots, src(\psi_m)) \cdot f(h_1[\chi_1, \ldots, \chi_m], \ldots, h_k[\chi_1, \ldots, \chi_m]) \cdot f(h_1[\phi_1, \ldots, \phi_m], \ldots, h_k[\phi_1, \ldots, \phi_m])$; cfr. Lem 5.3.9. Hence, an argument about $f(h_1[\chi_1, \ldots, \chi_m], \ldots, h_k[\chi_1, \ldots, \chi_m])$ analogous to that used in the infinitary multistep case for $f(\chi_1, \ldots, \chi_m)$, cfr. Lem. 5.3.9, allows to conclude.

  The other possible case is $h = x_j$, implying $\psi \approx_B \mu(src(\psi_1), \ldots, src(\psi_m)) \cdot \psi_j$. IH can be applied on ψ_j obtaining $\psi_j \approx_B \chi' \cdot \phi$, where χ' is a finite stepwise-or-nil proof term, ϕ is convergent and $mind(\phi) > n$. Thus we conclude by taking $\chi := \mu(src(\psi_1), \ldots, src(\psi_m)) \cdot \chi'$.

□

5.5.4 Proof of the compression result

Finally, we give the main result of this section.

Theorem 5.5.35. *Let ψ be a convergent proof term. Then there exists some stepwise proof term ϕ verifying $\psi \approx \phi$ and $steps(\phi) \leqslant \omega$.*

Proof. We define the sequences of proof terms $\langle\psi_i\rangle_{i<\omega}$ and $\langle\phi_i\rangle_{i<\omega}$ as follows. We start defining $\psi_0 := \psi$. Then, for each $i < \omega$, we define ϕ_i and ψ_{i+1} to be proof terms verifying that $\psi_i \approx_B \phi_i \cdot \psi_{i+1}$, ϕ_i is a finite stepwise-or-nil proof term and either $mind(\psi_{i+1}) > mind(\psi_i)$ or $mind(\psi_{i+1}) = mind(\psi_i) = \omega$; cfr. Prop. 5.5.34. Observe that $mind(\psi_i) < \omega$ implies $mind(\phi_i) = mind(\psi_i)$ by 5.3.6, so in that case ϕ_i is a stepwise proof term, i.e. it is not trivial. Moreover, an easy induction on n yields $\psi \approx_B \phi_0 \cdot \ldots \cdot \phi_n \cdot \psi_{n+1}$ for all n.

We define $T := \{n \ / \ \psi_n$ is a trivial proof term$\}$. There are three cases to consider:

- If $0 \in T$, i.e. if ψ is a trivial proof term, then it is enough to take $\phi := src(\psi)$ and refer to Lem. 5.3.10.

- Assume $0 \notin T$ and $T \neq \varnothing$, let n be the minimal element in T. In this case we take $\phi := \phi_0 \cdot \ldots \cdot \phi_{n-1}$. For any $k < \omega$, observe that $\psi \approx_B \phi \cdot \psi_n$, $\phi \approx_B \phi \cdot tgt(\phi)$ (cfr. $(\mathsf{IdRight})$), and $mind(\psi_n) = mind(tgt(\phi)) = \omega > k$, cfr. Lem. 5.2.11. Then Dfn. 5.3.3 allows to assert $\psi \approx \phi$. Finally, observe that each ϕ_i being finite implies that ϕ is also a finite stepwise proof term, i.e. it verifies $steps(\phi) < \omega$.

- Assume $T = \varnothing$. In this case, for any i Lem. 5.2.11 implies that $mind(\psi_i) < \omega$, so that ϕ_i is non-trivial. We take $\phi := \cdot_{i<\omega} \phi_i$. Let $n < \omega$. We have already verified that $\psi \approx_B \phi_0 \cdot \ldots \cdot \phi_n \cdot \psi_{n+1}$, and $\phi \approx_B \phi_0 \cdot \ldots \cdot \phi_n \cdot \cdot_{i<\omega} \phi_{n+1+i}$. On the other hand, an easy induction on k implies $mind(\psi_k) = mind(\phi_k) \geqslant k$ for all k, then $mind(\psi_{n+1}) > n$, and also $mind(\cdot_{i<\omega}\ \phi_{n+1+i}) > n$. Hence the rule Lim can be applied to obtain $\psi \approx \phi$. We conclude by observing that $steps(\phi_n) < \omega$ for all n implies that $steps(\phi) \leqslant \omega$.

□

Chapter 6

Conclusions

6.1 Rewriting systems: challenges, decisions and results

We deal in this thesis with rewriting systems of different sorts. In turn, a different subject is addressed in each case. The studied systems, and the main results presented for each one, can be summarised as follows.

Pattern calculi

> **Subject**: Normalisation.
>
> **Main result**: Definition of a normalising reduction strategy for the Pure Pattern Calculus (PPC).
>
> **Developed in**: Chapter 3.

Explicit substitution calculi

> **Subject**: Standardisation.
>
> **Main result**: Uniqueness of `s.r.s.`, modulo square equivalence, for the linear substitution calculus ($\lambda_{\mathtt{lsub}}^{\sim}$).
>
> **Developed in**: Chapter 4.

Infinitary, first-order term rewriting systems

> **Subject**: Equivalence of reductions.
>
> **Main result**: Characterisation of permutation equivalence by equational reasoning on proof terms.
>
> **Developed in**: Chapter 5.

We remark that pattern calculi, explicit substitution calculi, and infinitary rewriting, all emerged around 1990. We mention some of the earlier references in each case: [vO90] for pattern calculi, [HL89, ACCL91] for explicit substitution calculi, and [KKSdV90, DKP91] for infinitary rewriting. Hence, all these systems are considerably younger than the λ-calculus, and also than the study of generic properties and techniques for first-order term rewriting systems. While each of these families of rewriting systems has been a subject of considerable interest since its introduction, their features and formal properties are still not as well understood as those of more "classical" rewriting systems.

Moreover, in each case, there are features and peculiarities of the studied rewriting systems, which make the study of the chosen subjects a challenging task. We summarise these features, along with some choices made in this thesis.

6.1.1 Normalisation for the Pure Pattern Calculus

As described in Section 3.5, the failure mechanism of PPC implies that, to rewrite some terms to a normal form, it is not possible for a reduction strategy to select a single step, analysing only the term structure; that is, if for any set of terms having the same structure, the strategy always selects the step in the same position. E.g., consider

$$t = (\lambda_{\{x\}} \mathtt{a}\,\mathtt{b}\,\mathtt{c}\,x.\mathtt{b}\,\mathtt{d})(\mathtt{a}\,r_1 r_2 r_3)$$

where r_1, r_2 and r_3 are redexes. If r_1 rewrites to $\mathtt{d}$, then performing that step in t yields

$$t' = (\lambda_{\{x\}} \mathtt{a}\,\mathtt{b}\,\mathtt{c}\,x.\mathtt{b}\,\mathtt{d})(\mathtt{a}\,\mathtt{d}\,r_2 r_3)$$

which rewrites to I in one step, because the match of the pattern $\mathtt{a}\,\mathtt{b}\,\mathtt{c}\,x$ against the argument $\mathtt{a}\,\mathtt{d}\,r_2 r_3$ yields $\mathtt{fail}$. A similar situation holds if r_2 rewrites to $\mathtt{d}$. On the other hand, performing r_3 does not affect the status of the mentioned match; it is $\mathtt{wait}$ for any possible contractum. Consequently, in principle either of r_1 or r_2 could be selected to yield a normal form from t. But choosing r_1 would be a bad decision for a term u having the same shape as t, that is $u = (\lambda_{\{x\}} \mathtt{a}\,\mathtt{b}\,\mathtt{c}\,x.\mathtt{b}\,\mathtt{d})(\mathtt{a}\,r'_1 r'_2 r'_3)$, where r'_1 leads to an infinite reduction, whilst r'_2 rewrites to $\mathtt{d}$. An analogous reasoning invalidates the selection of r_2.

The reduction strategy defined in Section 3.5 chooses both steps in t. Therefore, it is a *multistep* (cfr. Section 3.1.1) reduction strategy.

The described behavior of PPC is related with the notion of *non-sequentiality*, and also with the fact that PPC does not enjoy the ARS Stability axiom; cfr. Section 1.4.1, the introduction to Chapter 3, and Section 3.4.2.

6.1.2 Standardisation for the Linear Substitution Calculus

The reduction space of several of the explicit substitution calculi known by the author are extremely complex, due to the multiplicity of possible interleavings between β-steps and explicit substitutions, and to the interplay between different explicit substitutions. While one of the aims for the introduction of explicit substitution calculi *at a distance* is to obtain simpler reduction spaces, that of $\lambda_{\mathtt{lsub}}^{\sim}$ is still more complex than that of the λ-calculus, as an unavoidable consequence of the inclusion of the substitution operation inside the language.

Furthermore, a characterisation of standard reduction sequences based on the notion of *external* steps, as described in Section 1.3.1 and Section 2.1.8, requires an order to be defined on coinitial steps; this order is the embedding relation included in the ARS model. In λ-calculus, the embedding is closely related with the syntactic nesting of redexes. This is not the case for $\lambda_{\mathtt{lsub}}^{\sim}$, implying a more complex definition of the embedding. There are two reasons for this gap between semantic embedding and syntactic nesting:

1. The *graphical equivalence* introduced in Section 4.1 and noted $\sim$, which relates different terms describing the same *linear logic proof-net* [Gir87]. While the embedding between steps should be invariant w.r.t. $\sim$-equivalence, the latter equates terms having different structure. E.g., consider the following equivalent terms

$$\begin{array}{ccccc} t_1 & & t_2 & & t_3 \\ (y(xr))[x/3][y/4] & \sim & (y(xr))[y/4][x/3] & \sim & (y[y/4](xr))[x/3] \end{array}$$

where r is a redex. Each of these terms includes three steps: two `ls`-steps, for the occurrences of x and y bound by the explicit substitutions $[x/3]$ and $[y/4]$ respectively, and the step whose pattern is r. Let us call these steps a_x, a_y and b respectively. In t_1, the subterm corresponding to a_x is $(y(xr))[x/3]$, while a_y involves the whole term. Hence we have a_y nesting a_x in t_1. A similar analysis yields that a_x nests a_y in t_2. In turn, in t_2 we have a_y nesting b, whilst these two steps are disjoint in t_3.

2. The fact that a step a can duplicate another step b, where a does not syntactically nest b. E.g., let us consider

$$t = ((x[x/y])(w[w/2]))[y/4]$$

and call again a_x and a_y the steps corresponding to the occurrences of x and y bound by the corresponding explicit substitutions. Performing a_x yields

$$t' = ((y[x/y])(w[w/2]))[y/4]$$

thus duplicating a_y; observe the two occurrences of y bound by $[y/4]$ in t'. But a_x does not nest a_y; in fact, it is the other way around.

We introduce an embedding relation, the *box order* defined in Section 4.5, which takes into account these considerations. The resulting ARS does not enjoy the Enclave–Creation nor the Enclave–Embedding axioms, reflecting the subtle behavior of $\lambda^{\sim}_{\mathtt{lsub}}$; cfr. Section 4.5.2.

The desired standardisation results are obtained by developing a novel abstract proof in the ARS model, described in Section 4.6.

6.1.3 Equivalence of reductions in infinitary rewriting

As described in Section 1.4.3, equivalence of reductions is defined as the set of equations on *proof terms* deducible by equational logic from a set of basic equations. In turn, the basic equations describe the permutation of adjacent steps. The result is the *permutation equivalence* relation $\approx$ on proof terms.

Infinity arises in the equivalence judgements of the defined equational theory in several ways, besides the basic fact that proof terms can be infinite objects. We comment some sources of infinity in these judgements, using the rules $\mu : f(x) \to g(x)$ and $\nu : g(x) \to j(x)$.

- Consider the reduction sequences:

$$\begin{array}{l} f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega \to j(g^\omega) \to j(j(g^\omega)) \twoheadrightarrow j^\omega \\ f^\omega \to g(f^\omega) \to j(f^\omega) \to j(g(f^\omega)) \to j(j(f^\omega)) \twoheadrightarrow j^\omega \end{array}$$

corresponding to the proof terms

$$\cdot_{i<\omega}\ g^i(\mu(f^\omega)) \cdot \cdot_{i<\omega}\ j^i(\nu(g^\omega)) \qquad\qquad \cdot_{i<\omega}\ j^i(\mu(f^\omega)) \cdot j^i(\nu(f^\omega))$$

Both reduction sequences comprise exactly the transformation of each f into g, and subsequently into j. The difference between them is the *order* in which the steps are performed. Hence the reduction sequences are permutation equivalent, so that the same should happen to the proof terms denoting them.

Observe that to transform the former reduction sequence, or equivalently the former proof term, into the latter one, an *infinite* number of ν-steps must be permuted, each one with an *infinite* number of μ-steps.

- Consider the terms described by the equations

$$t = h(f(f(a)), t) \qquad\qquad t' = h(g(g(a)), t')$$

and the following two reduction sequences which transform t into t': one which applies successively the two steps $f(f(a)) \to f(g(a)) \to g(g(a))$ in each occurrence of $f(f(a))$ in t, and the other analogously for the steps $f(f(a)) \to g(f(a)) \to g(g(a))$. To verify the equivalence of these infinite reduction sequences, an infinite number of disjoint step permutations must be considered. From a different point of view, each of these reduction sequences is an infinite concatenation where each component is the two-step sequence shown; a step permutation must be performed on *each of the infinite components* to verify their equivalence.

Both issues suggest the need to define some kind of "getting to limits", cfr. [Kah10], in order to build equational reasonings of an infinite nature. Furthermore, some method to reason about the set of valid judgements must be provided. Length, and also depth, of judgements turn out to be inadequate in the infinitary setting. We remark that equational reasoning on infinite objects is a subject of current research; cfr. [Kah10].

We handle these issues by adding the following features to the equational logic-based definition of the permutation equivalence relation $\approx$.

1. We add a specific equational logic rule to handle step permutations in each component of an infinite concatenation.

2. We add a rule to formalise the idea of "getting to limits": given two reduction sequences ψ and ϕ, if for any $\epsilon > 0$, ψ can be proved equivalent to some ψ', and ϕ equivalent to some ϕ', such that the distance between ψ' and ϕ' is less than ϵ, then we can conclude that ψ and ϕ are themselves equivalent. The distance between two strongly convergent reduction sequences is inverse to the minimum depth of their respective remaining parts following a maximal common prefix. We notice that the "getting to limits" rule cannot be used, in turn, when proving $\psi \approx \psi'$ and $\phi \approx \phi'$.

3. We associate a countable ordinal to each judgement, so that the ordinal for the conclusion of a rule is strictly greater than that of any of its premises. In this way, transfinite induction can be used to reason about judgements, even in the presence of rules having an infinite number of premises.

6.2 Generic models in this thesis

The material and achievements of this thesis can be seen from a different perspective: that of the *generic models* used to study the addressed rewriting systems. In the author's opinion, the material included in this thesis supports the utility and value of the generic models as a tool to study rewriting systems.

We remark that, as discussed in Section 6.1, the study of the subjects (normalisation for PPC, standardisation for $\lambda_{\mathtt{lsub}}^{\sim}$, equivalence of reductions for infinitary rewriting) we examine through the ARS and proof term models, is a far from trivial task. We also want to point out that we were able to *extend* the presentations of the ARS and proof term models taken as reference ([Mel96] and [BKdV03] Sections 8.2 and 8.3, respectively). Such extensions are needed in order to obtain the desired results about the rewriting systems concerned in this thesis, summarised in page 213. In all the cases, the extensions preserve the basic ideas underlying the conception of each generic model.

In the rest of this section, we examine the conclusions we can draw from the use of generic models in this thesis, in different aspects:

- the insights that the ARS and the proof term models can give to the understanding of the behavior of a rewriting system,
- the contributions made in this thesis to the ARS and the proof term models, related to the extensions to these models just commented, and
- some preliminary notes for a comparison between different generic tools to study rewriting systems.

6.2.1 Generic models give useful insights

The ARS model

The Pure Pattern Calculus (PPC) and the Linear Substitution Calculus ($\lambda_{\mathtt{lsub}}^{\sim}$) are modeled in this thesis as Abstract Rewrite Systems (ARS).

As a result from this experience, we found that the ARS model gives useful insights to the understanding of a rewriting system's behavior. This seems to be particularly the case for the *embedding axioms*, described in Section 2.1.5. The failure of a system to uphold an axiom indicates peculiarities to be taken into account when assessing that system's properties. On the contrary, the *axioms being verified by a rewriting system* indicate aspects in which the behavior of that system coincides with what is intuitively expected.

The Linearity axiom expresses a basic condition a step must satisfy to have the power to duplicate or erase other steps. It is a basic guide to shape the embedding relation, as described for $\lambda_{\mathtt{lsub}}^{\sim}$ in Section 4.5. This axiom, together with Context-Freeness, form a basic regularity requirement for the residual and embedding relations. All the rewriting systems studied in this thesis (namely PPC, $\lambda_{\mathtt{lsub}}^{\sim}$ and the first-order, left-linear, infinitary term rewriting systems) satisfy both axioms.

We notice the difference between the statement of the Context-Freeness axiom as it appears in [Mel96] and in this thesis, and the following stronger variant, proposed in [GLM92]:

Let $a, b, c \in \mathcal{RO}(t)$, $b[\![a]\!]b'$, and $c[\![a]\!]c'$.
Then $a < c$, or $(b < c \leftrightarrow b' < c') \wedge (c < b \leftrightarrow c' < b')$.

This statement does not hold, neither for the λ-calculus if we consider the argument order (i.e., that the α-labeled step in $(\lambda x^\alpha.s)u$ embeds the steps inside u, and does not embed those inside s), nor for $\lambda^{\sim}_{\mathtt{lsub}}$ if we consider the analogous box order. A simple example for each of these rewriting systems follow:

$$(\lambda x^\alpha.I^\gamma x)r^\beta \xrightarrow{a} I^\gamma r^\beta \qquad\qquad (\lambda x^\alpha.I^\gamma x)r^\beta \xrightarrow{a} I^\gamma x\,[x/r^\beta]$$

where a, b and c are the steps labeled with α, β and γ respectively, in the term $(\lambda x^\alpha.I^\gamma x)r^\beta$. In both cases, $a \not< c$, $c \not< b$ and $c' < b'$.[1] The weaker version of Context-Freeness allows to use the box order for $\lambda^{\sim}_{\mathtt{lsub}}$.

Besides Linearity and Context-Freeness, PPC enjoys the Enclave–Creation and Enclave–Embedding axioms, and it does not satisfy Stability. The counterexamples for the latter correspond to disjoint ways of creating a *matching failure*, which is precisely the characteristic which must be particularly considered in order to define a normalising reduction strategy for that calculus.

On the other hand, the fact that PPC enjoys both *enclave* axioms, as well as Context-Freeness, yields a high degree of invariance in the embeddings of residuals. This observation is further reinforced by the addition of the novel axiom Pivot, which complements Context-Freeness, Enclave–Embedding and Enclave–Creation; cfr. Section 3.1.4. The degree of invariance stated by these axioms is required to verify that a certain kind of *multistep permutation* is allowed (namely, that it is possible to permute a *dominated* multireduction w.r.t. a *free* one, such that the dominated and free properties are unaltered), cfr. Section 3.3.2. In turn, the latter result is crucial in order to prove that the defined strategy is normalising.

The situation for $\lambda^{\sim}_{\mathtt{lsub}}$ is somewhat inverse: besides Linearity and Context-Freeness, while we conjecture that this calculus, equipped with the box order, enjoys Stability, we show through counterexamples that it does not satisfy neither Enclave–Creation nor Enclave–Embedding; cfr. Section 4.5.2. We also conjecture that $\lambda^{\sim}_{\mathtt{lsub}}$ enjoys the following weakened forms of Enclave–Creation and Enclave–Embedding.

Enclave–Creation (−) Let $b < a$, $b[\![a]\!]b'$ and $\varnothing[\![a]\!]c'$. Then $c' \not< b'$.

Enclave–Embedding (−) Let $b < a < c$, $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Then $c' \not< b'$.

indicating that the irregular behavior of this calculus is *bounded* in some sense: if $b < c$, then c' can be free from (i.e. not embedded by) b', but it cannot embed b' in turn.

A note on Stability: as shown in Section 4.5.2, there is a case in which a step (again, a gc-step) can be created by the contraction of either of two different steps. But Stability is not compromised, since the creating steps are not disjoint.

The proof term model

Reduction sequences, or more generally any form of *contraction activity* (cfr. Section 1.3.2), for infinitary first-order, left-linear term rewriting systems, and also the equivalence on reduction sequences (in fact, on any form of contraction activity), are described in the proof term model.

In our opinion, this representation contributes to the understanding of some aspects in infinitary rewriting.

[1] In the $\lambda^{\sim}_{\mathtt{lsub}}$ example, $c' \prec^2_{\mathtt{B}} b'$, since $c' \prec^1_{\mathtt{B}} d' \prec^1_{\mathtt{B}} b'$ where d' is the created ls-step on the free occurrence of x in $I^\gamma x\,[x/r^\beta]$.

- The characterisation of permutation equivalence yields a clear view about the nature of the permutations needed to prove the equivalence of two reduction sequences. It also helps to understand in which cases some contraction activity exploiting parallelism, i.e. a simultaneous development, and a reduction sequence (that is, a totally sequential form of contraction activity), are equivalent in the infinitary setting.

- The obtained characterisation of permutation equivalence allows to shed some light on the phenomenon of *infinitary erasure*, cfr. Section 5.3.4.

- Proof terms can also be a tool to analyse *infinitary developments*, which correspond exactly to the base layer in the definition of infinitary proof terms given in Sections 5.2.1 and 5.2.2. This direction is not pursued in the present thesis; we describe some conjectures in Section 5.5.1, cfr. a footnote in page 196. In the literature, cfr. [BKdV03], Section 12.5, fore a study of properties of infinitary development using different tools.

6.2.2 Contributions to the generic models

The ARS model

We summarise the main results obtained by means of the ARS model in this thesis: for PPC, we prove that the reduction strategy defined is indeed normalising; and w.r.t. $\lambda_{\mathtt{lsub}}^{\sim}$ we prove, for every reduction sequence, the existence of an equivalent `s.r.s.`, and moreover that such `s.r.s.` is unique modulo square equivalence.

Similar results are present in [Mel96]. In that work, the normalisation of *external* reduction strategies is proved in Section 5.2 (cfr. Thm. 5.2, page 137), and the existence and uniqueness (modulo square equivalence) of `s.r.s.` for any class of equivalent reductions is stated in Section 4.3 (cfr. Thm. 4.2, page 81) and proved in Section 4.4. However, both of these results require the ARS to satisfy *all* the embedding axioms, which is not the case for neither PPC nor $\lambda_{\mathtt{lsub}}^{\sim}$ (endowed with the box order); cfr. Sections 3.4.2 and 4.5.2.

This thesis includes two novel proofs, which allow to obtain normalisation and standardisation results for ARS which fail to verify some of the embedding axioms.

1. The novel *normalisation* proof is described in Section 3.3, cfr. Thm. 3.3.14, and applied to PPC in Section 3.5.2, cfr. Thm. 3.5.26.

2. The *standardisation* proof is described, and applied to $\lambda_{\mathtt{lsub}}^{\sim}$, in Section 4.6, cfr. Thm. 4.6.3 and Thm. 4.6.5.

We think that these proofs can be considered as a contribution to the ARS model, and specifically to its capability to yield strong results for rewriting systems showing peculiarities in their behavior.

We remark that the *abstract normalisation proof* included in this thesis involves the use of the gripping relation and the concepts of multisteps and multireductions, elements of the ARS model present in [Mel96], but applied there to other purposes. Namely, the gripping relation is introduced as a means to state and prove an abstract proof of FD,[2]

[2]which we use to obtain FD for PPC, cfr. Section 3.4.3, page 82.

while multisteps and multireductions are applied in an abstract proof of confluence; cfr. Chapter 3 and Sections 2.3.2 to 2.3.6 in [Mel96], respectively.

On the other hand, the *standardisation proof* makes a subtle use of *two* ARS, differing only in their embedding relation. The morale of this proof can be described as follows:

> If an ARS $\mathfrak{A}_P$, whose embedding relation is a partial order, verifies the initial axioms, FD, SO, Linearity and Context-Freeness, and this partial order can be completed to a total order, resulting in an ARS $\mathfrak{A}_T$, such that $\mathfrak{A}_T$ verifies the remaining embedding axioms (namely, Enclave–Creation, Enclave–Embedding and Stability) as well, so that $\mathfrak{A}_P$ enjoys the existence of `s.r.s.` result and $\mathfrak{A}_T$ satisfies the stronger uniqueness of `s.r.s.` result, then the uniqueness result can also be obtained for $\mathfrak{A}_P$.

We remark that this argument does not involve the Stability axiom. For the original ARS, satisfying this axiom is not required. For the ARS whose embedding relation is a total order, there are no a, b verifying $a \parallel b$, while the statement of Stability, cfr. Section 2.1.5, has the form "Assume $a \parallel b$. Then ...". Therefore, the axiom holds immediately for such ARS.

One could wonder whether this statement applies to the ARS defined for PPC in Section 3.4.3, obtaining in this way a result of uniqueness of `s.r.s.` for that rewriting system. The answer is negative, because there is no way to complete the embedding relation to a total order such that the resulting ARS verifies the Enclave–Creation axiom. Let us consider the term

$$t = (\lambda_{\{x\}} \mathtt{p}\,\widehat{x}\,\mathtt{m}\,\mathtt{s}.x)\ (\mathtt{p}\,(\underbrace{I\mathtt{a}}_{a})(\underbrace{I\mathtt{f}}_{b})(\underbrace{I\mathtt{d}}_{c}))$$

whose three steps are indicated. As described in Section 3.4.2, we have

$$t \xrightarrow{b} \overbrace{(\lambda_{\{x\}} \mathtt{p}\,\widehat{x}\,\mathtt{m}\,\mathtt{s}.x)\ (\mathtt{p}\,(\underbrace{I\mathtt{a}}_{a'})\,\mathtt{f}\,(\underbrace{I\mathtt{d}}_{c'}))}^{d'} \qquad t \xrightarrow{c} \overbrace{(\lambda_{\{x\}} \mathtt{p}\,\widehat{x}\,\mathtt{m}\,\mathtt{s}.x)\ (\mathtt{p}\,(\underbrace{I\mathtt{a}}_{a''})(\underbrace{I\mathtt{f}}_{b''})\,\mathtt{d})}^{d''}$$

where $c[\![b]\!]c'$, $b[\![c]\!]b'$, $\varnothing[\![b]\!]d'$, $\varnothing[\![c]\!]d''$, $d' < c'$ and $d'' < b''$. Let $<_t$ be a total order extending $<$, so that $d' <_t c'$ and $d'' <_t b''$. Moreover, b and c must be comparable w.r.t. $<_t$. If $b <_t c$, then $d'' <_t b''$ contradicts Enclave–Creation. Otherwise $c <_t b$, in this case the axiom is contradicted by $d' <_t c'$.

The proof term model

The main, and obvious, contribution to the proof term model is its extension to infinitary (first-order, left-linear) term rewriting. We remark that, as verified in Section 5.4, any reduction sequence[3] can be denoted by a proof term, and moreover that this denotation is *unique* modulo rebracketing.

The definition of the set of proof terms can be considered as an extension of that given for the finitary case in [BKdV03], Section 8.2. The basic principles are the same: the signatures coincide, the restrictions for a term in that signature to be a valid proof term apply to the occurrences of the dot, i.e. the concatenation symbol. The main differences

[3]More precisely, any reduction sequence whose length is a countable ordinal, therefore including particularly all convergent reduction sequences. Cfr. [KdV05].

are the addition of a *convergence condition* to these restrictions, and a special formation rule for infinite concatenations.[4]

Analogously, the characterisation of equivalence on reductions extends that given for finitary proof terms. Again the basic idea, namely to model permutation equivalence by resorting to equational logic, coincide. Moreover, the basic equations are the same. The additions for infinitary rewriting are those detailed in Section 6.1.3: special care is taken with infinite concatenation, a rule is added to model the "getting to limits" operation, and transfinite induction is used to reason about equivalence judgements. We point out that the characterisation of the equivalence of reductions given in [BKdV03] Section 8.3 fail to capture all the cases of reduction sequences which sanctioned as equivalent in infinitary rewriting, particularly those involving infinite reduction sequences. The extension we introduce in Section 5.3 addresses these shortcomings; several examples are given there.

In our opinion, the fact that the main definitions of the proof term model can be extended to infinitary rewriting, preserving the ideas underlying those definitions, is an argument in favor of the strength of this generic model of rewriting systems. We notice that the compression proof presented in Section 5.5 shows the capability of the obtained model of infinitary rewriting to develop proofs of relevant properties.

On another front, we want to stress that the characterisation of permutation equivalence given in Section 5.3.2 is a successful case of equational logic applied to infinitary objects. We hope this work contributes to the development of infinitary equational logic on its own.

6.2.3 Towards a comparison of generic models

We want to remark some points about the nature, features and strengths of the two generic models used in this thesis. Of course, the following comments reflect the author's personal view, obtained from the experience of the work which led to this thesis.

The strengths of each model stem, in the author's opinion, from the principles shaping each of them.

The **ARS model** has a *highly abstract* nature, as highlighted in its name. We recall that all the information about the steps is given in the form of relations (such as the residual, embedding and gripping relations described in this thesis); besides them, only the identity of each step, and its source and target objects, are part of the model. The axioms modeling the expected features of a rewriting system are defined in terms of these abstract relations. In turn, the statement of the properties which can be proved in this model refer to the relations and axioms just mentioned.

In the author's view, the modeling of rewriting systems as ARS offers a remarkable value, besides the possibility of profiting from the abstract proofs which can be expressed in that model. The abstract nature of the elements comprising an ARS (cfr. Section 2.1.1) and the description of the features of rewriting systems given by the axioms (cfr. Section 2.1.3 to 2.1.6), provide a framework which can give useful insights for the better understanding of a system's behavior, as described for PPC and $\lambda_{\mathtt{lsub}}^{\sim}$ in

[4] While the finitary and infinitary definitions of proof terms differ in one aspect, namely how the source and target of a proof term are computed (cfr. Section 2.2.2 and Sections 5.2.1, 5.2.2 for finitary and infinitary proof terms respectively), the infinitary definitions are perfectly adequate for the finitary proof terms as well.

Section 6.2.1, and which can also be valuable for the comparison of rewriting systems having different conceptions.

The **proof term model**, in turn, is limited to term rewriting. In this model, the structure of the terms being rewritten is preserved in the proof terms. A consequence is that this model must be explicitly adapted in order to expand its scope to rewriting systems differing in their basic features, as done in this thesis for infinitary rewriting, and in [Bru08] for higher-order rewriting.

The proof term model is focused on individual proof terms, where each proof term denotes a particular reduction. The identity of a step, which is a central concept in the ARS model, is not present in the proof term model. The detailed description of a particular reduction in this model, allows to denote, distinguish, and also combine, *sequential* and *parallel* contraction. *Localised* contraction can also be adequately described in the proof term model; cfr. Section 1.3.2. Moreover, in the author's opinion, the explicit rendering given by proof terms yields a more descriptive view of how a reduction can be transformed in another, equivalent one. Considering a first-order term rewriting system including the rule $\mu : f(x) \to g(x)$, the equivalence of the reduction sequences

$$f(f(a)) \to f(g(a)) \to g(g(a)) \qquad \text{and} \qquad f(f(a)) \to g(f(a)) \to g(g(a))$$

can be justified in the proof term model by means of the following permutation equivalence judgement

$$f(\mu(a)) \cdot \mu(g(a)) \approx \mu(\mu(a)) \approx \mu(f(a)) \cdot g(\mu(a))$$

In the author's opinion, this justification is a clear description of the permutation of steps which allows to transform the first reduction sequence into the second, or vice versa; furthermore, it also suggests that both steps could also be contracted simultaneously, yielding a third option to organise the contraction of these steps.

The rendering of the same situation in the ARS model is somewhat less direct, in the author's view. We could have the steps

$$\begin{array}{llll} a: & f(f(a)) \to g(f(a)) & b: & f(f(a)) \to f(g(a)) \\ a': & f(g(a)) \to g(g(a)) & b': & g(f(a)) \to g(g(a)) \end{array}$$

and $a[\![b]\!]a', b[\![a]\!]b'$ in the residual relation, so that the definition of permutation equivalence in the ARS model, cfr. Dfn. 2.1.18, yields $b; a' \approx a; b'$.

The nature of each model is reflected in the way **orthogonality** is handled in each of them. Recall that from a syntactic perspective, orthogonality is related with the absence of *critical pairs* [KB70, Hue80]. Local confluence of orthogonal steps arises as a consequence. This view leads to weak forms of orthogonality, such as the definition of *almost orthogonal* and *weakly orthogonal* rewriting systems; cfr. [vO94, vR97, vO99].

The *ARS model* defines orthogonality from a different perspective: two steps are *defined as* orthogonal iff the contraction of their respective residuals yields a local confluence diagram. This is the Semantic Orthogonality, or SO, axiom, introduced in Section 2.1.4. Taking into account this semantical perspective of orthogonality, we can consider some almost orthogonal and weakly orthogonal rewriting systems in equal terms with orthogonal ones.[5]

[5]However, this is not always the case. Consider the first-order term rewriting system whose only rule

In this thesis, we have profited from the abstract view of orthogonality given by the ARS model. The complex matching operation of PPC weakens the meaningfulness of the concept of critical pair to understand this calculus. In turn, $\lambda^{\sim}_{\texttt{lsub}}$ has critical pairs. Nevertheless, we consider PPC and $\lambda^{\sim}_{\texttt{lsub}}$ as *semantically orthogonal* systems.

We mention that systems which fail to satisfy the semantic characterisation of orthogonality, can also be studied in the ARS model. A *compatibility* relation is added to identify the pairs of mutually orthogonal steps. We did not use this relation in the present thesis.

On the other hand, a *proof term* denotes a particular reduction, in which for any potential conflict due to a critical pair, a decision has already been taken. Proof terms encode explicitly each decision taken: different decisions for the same critical pair give rise to different, non equivalent, proof terms. E.g., if we consider the rules

$$\mu : f(x) \to g(x) \qquad \tau : f(j(x)) \to g(n(x)) \qquad \pi : j(x) \to n(x)$$

then the proof terms $\mu(\pi(a))$ and $\tau(a)$ are not equivalent, even when their source and target terms coincide, $f(j(a))$ and $g(n(a))$ respectively. While the proof term $\mu(\pi(a))$ can be proved to be equivalent to either sequentialisation of its two steps, i.e. $\mu(j(a)) \cdot g(\pi(a))$ and $f(\pi(a)) \cdot \mu(n(a))$, neither of these proof terms can be equated, by means of the equations and rules described in Section 5.3.2, to $\tau(a)$.[6]

In the general case, assume two steps a and b, which form a critical pair. There is no way to contract a, and subsequently a residual of b, in the same reduction sequence. Analogously, it is not possible to contract b and later a residual of a. The reason for this is straightforward: the contraction of a leaves no residuals of b, and vice versa. As proof terms denote reduction sequences,[7] no proof term can be built in which the contraction of both a and b (or their residuals) is denoted. The focus that the proof term model puts on individual reductions, implies that lack of orthogonality of a rewriting system is simply not a concern.

Finally, we want to mention that the **generic formalisms** for the definition of higher-order rewriting systems, such as *CRS* [Klo80], *HRS* [Nip91, MN98] and *ERS* [GKK05], can be considered as generic models for the study of rewriting systems, just as the ARS and proof term models used in this thesis.

If a given rewriting system, let us call it A, can be adequately modeled in one of these formalisms, then (obviously) any statement proved for that formalism is automatically valid for A (provided that the corresponding hypotheses hold). Such model can also give

is $f(f(x)) \to x$. Then the term $f(f(f(f(a))))$ has three steps, according to the following diagram:

$$\overbrace{f\,(\underbrace{f\,\overbrace{(f(f(a)))}^{c}}_{b})}^{a}$$

In this case, there exist a', c' verifying $a[\![c]\!]a'$ and $c[\![a]\!]c'$. On the other hand, $a[\![b]\!] = c[\![b]\!] = b[\![a]\!] = b[\![c]\!] = \varnothing$. Therefore $c[\![a; b[\![a]\!]]\!]c'$ while $c[\![b; a[\![b]\!]]\!] = \varnothing$, thus breaking SO.

An analysis of such cases of redex overlapping could benefit from the notion of *weakly orthogonal projection*, as defined in e.g. [KKvO04] Section 2.4, and also from the material given in [BKdV00] Section 5, where the notion of *cluster residual* is introduced.

[6]This is also the case for the equivalence relation on proof terms which characterise permutation equivalence for finitary rewriting, described in this thesis in Section 2.2.3

[7]In fact, proof terms can denote different forms of *contraction activity*, where reduction sequences are one of these forms.

valid insights for the understanding of the features of A, and for possible comparisons with other systems.

As shown in Chapter 4 for $\lambda^{\sim}_{\mathtt{lsub}}$, to render a rewriting system in the ARS model, and to prove that some expected feature holds for the resulting ARS, can be a difficult task. On the other hand, we remark that the situation is analogous for the generic formalisms for higher-order calculi: to describe a given rewriting system in one of these formalisms can be far from trivial as well. We notice that the Pure Pattern Calculus has been recently [vOvR14] modeled using the generic *HRS* formalism [Nip91, MN98].

6.3 Further work

We briefly describe possible lines of further investigation for each of the three main directions developed in this thesis.

Normalisation for pattern calculi

We recall the two main results obtained in this direction: the definition of the multistep reduction strategy for PPC we called $\mathcal{S}$, cfr. Section 3.5, and an abstract normalisation proof developed in the ARS model, which we used to verify that the strategy $\mathcal{S}$ is normalising, cfr. Section 3.3.

In our opinion, the scope of the work presented in this thesis can be expanded in both aspects, i.e. the definition of reduction strategies and the normalisation proof.

To this effect, a possible research direction is to elucidate whether the ideas underlying the definition of $\mathcal{S}$ can lead to the definition of strategies for other rewriting systems, and whether the eventually obtained strategies can be proven normalising by resorting to the abstract normalisation proof described in this thesis. Particularly, it would be interesting to obtain families of calculi definable in some generic formalism for higher-order term rewriting systems, such as HRS, CRS or ERS, cfr. Section 6.2.3, for which positive results in this direction could be obtained.

In a different direction, possible extensions or variations of the normalisation proof can be analysed, with the aim to broaden its scope. Particularly, we notice that the proof is applied in this thesis to the strategy $\mathcal{S}$, which selects always a subset of the set of *outermost* steps, in a term. On the other hand, the proof does not apply, in its present form, to the *parallel-outermost* reduction strategy, which indicates the simultaneous reduction of *all* the outermost steps in any term. This unpleasant observation is due to the fact that the non-gripping property of a set of steps is not necessarily preserved in its supersets. Specifically, if we call $\mathcal{O}(t)$ the set of outermost steps in the term t, then the set $\mathcal{O}(t)$ does not satisfies the non-gripping property in the general case. E.g., consider the term

$$t = (\lambda_{\{x\}}\mathtt{a}\,x.\,\underbrace{Dx}_{b})(\underbrace{I(\mathtt{a}\,\mathtt{b})}_{a})$$

whose only steps are a and b. As both steps are outermost, we have $\mathcal{O}(t) = \{a, b\}$. Contracting a results in

$$\overbrace{(\lambda_{\{x\}}\mathtt{a}\,x.\,\underbrace{Dx}_{b'})(\mathtt{a}\,\mathtt{b})}^{c'}$$

where $b[\![a]\!]b'$ and $c' \ll b'$. Hence $\mathcal{O}(t)$ does not enjoy the non-gripping property. On the other hand, $\mathcal{S}(t) = \{a\}$, this set is indeed non-gripping; observe $\mathcal{S}(t) \subset \mathcal{O}(t)$.

We conjecture that some variation of the given proof could apply to the parallel-outermost strategy, e.g. for PPC. In this perspective, it could be possible that the property of always selecting *necessary* sets of steps could suffice to guarantee that a reduction strategy is normalising. A proof of this conjecture, or a counterexample falsifying it, would be an interesting result in this direction.

Besides the proposed initiatives, which are of a theoretic nature, another avenue for further work is to test the practical feasibility of the strategy $\mathcal{S}$ by developing an interpreter of PPC based on it.

Standardisation for explicit substitution calculi

A possible direction of future work is the application of the ideas underlying the conception of the ES calculi *at a distance*, and the standardisation results for $\lambda^{\sim}_{\texttt{lsub}}$ presented in Chapter 4, to the study of the phenomenon of pattern matching, through the definition of *calculi with explicit matching* for which standardisation results can be stated.

Several calculi with explicit matching, inspired from pattern calculi, have been proposed; cfr. [For02, CK04, dCPdF11], and particularly [Bal10a] where a proposal based on PPC is presented. While several properties, including confluence, simulation, preservation of strong normalisation, and also properties of typed versions, have been stated for these calculi, the author is aware of no result about standardisation for calculi with explicit matching.

We conjecture that the application of the idea of *distant substitution*, expressed in the ls-rule of the $\lambda^{\sim}_{\texttt{lsub}}$, namely $C[\![x]\!]\,[x/u] \;\to\; C[\![u]\!]\,[x/u]$, can lead to the definition of calculi with explicit matching having simpler reduction spaces, as described for $\lambda^{\sim}_{\texttt{lsub}}$ in Section 1.2.2.

Another interesting aspect derived from the material in Chapter 4 is the characterisation of s.r.s. given by the *box order*, defined in Section 4.5. Based on the isomorphism between $\lambda^{\sim}_{\texttt{lsub}}$ and linear logic proof-nets given by the *graphical equivalence* relation $\sim$ on $\lambda^{\sim}_{\texttt{lsub}}$ terms, and the fact that the standardisation results described in Section 4.6 are stable by $\sim$, we conclude that the criterion described by the box order gives a sound standardness notion for proof-nets. The possible application of analogous criteria to other graph rewriting systems can be a subject of further investigation. Moreover, we mention the existence of a different notion of standardisation for proof-nets, namely the *standardisation by levels*, or depths [dCPdF11]. While the respective orders on coinitial steps are in general incomparable, the study of possible relations between them could shed some additional light about the behavior of proof-nets and ES calculi.

Infinitary rewriting

Some of the ideas underlying the proof of the compression result described in Section 5.5, can be extended in order to obtain *standardisation* results for infinitary rewriting. As noted in [Ket12], a concept of standard reduction being adequate for infinitary rewriting should be used, since leftmost-outermost reduction does not fit in this setting. In terms of the ARS model, an adequate *embedding relation* is needed. In our opinion, it is possible to prove the existence of a *unique* standard reduction in each permutation equivalence class, using depth-leftmost standardness as defined in [Ket12].

In turn, we hope that the eventual standardisation proof obtained can be applied to the finitary case as well, thus yielding a standardisation proof, based on proof terms, for first-order, left-linear finitary term rewriting. This proof would be an alternative to that presented in [BKdV03], Section 8.5.

Another avenue of further research on infinitary rewriting, for which the proof term model can be adequate, is the comparison of permutation equivalence, as defined in Section 5.3, with other notions of equivalence of reductions. We notice that in [BKdV03] Chapter 8, and also in [vOdV02], the equivalence of several such notions is established for finitary rewriting. We observe also that equivalence of infinitary reductions is defined in [KKSdV95] and [BKdV03] Chapter 12, by extending the Lévy equivalence criterion [HL91], based on the projection of reductions.

We also mention the possibility of using proof terms (more precisely, infinitary multisteps) to study properties of infinitary developments, as suggested in Section 6.2.1.

With respect to the extension of the proof term model to infinitary rewriting, a variant of the equational logic defined to model permutation equivalence, in which the Lim-rule can be used at most once in a derivation, and only as its last step, is worth considering. We notice that the derivations of permutation equivalence in our examples in Sections 5.3.3 and 5.3.4 are all of this form. We conjecture that a proof-theoretic analysis could yield the equivalence between this restricted variant and the more general version defined in this thesis.

Appendix A

Resumen en castellano

A.1 Introducción

La teoría de la **reescritura** es el estudio de la transformaciones discretas y paulatinas de cualesquiera objetos. Si los objetos de las transformaciones estudiadas son *términos*, es decir, cadenas bien formadas de símbolos, entonces se habla de **reescritura de términos**.

La teoría de la reescritura influye, en forma significativa y sostenida en el tiempo, en diferentes áreas dentro de la ciencia de la computación. Respecto de la *teoría de la computación*, destacamos que el *cálculo-lambda*, uno de los sistemas de reescritura más antiguos y más extensamente estudiados, conforma un modelo de cómputo equivalente a los basados en máquinas de Turing y en funciones recursivas. En relación con la *programación informática*, probablemente la contribución más relevante de la teoría de la reescritura es el rol preponderante que tuvo el cálculo-lambda para el surgimiento del *modelo funcional de la programación*. La influencia del modelo funcional en la comunidad global de programación está en rápido aumento, lo que se manifiesta tanto por la popularidad creciente de *lenguajes funcionales*, basados preponderantemente en este modelo, como por la adopción de técnicas y conceptos surgidos en el modelo funcional, en otros lenguajes de programación, así como en la práctica de profesionales que no necesariamente utilizan los lenguajes funcionales recién mencionados.

Un ejemplo sencillo de reescritura es la simplificación de expresiones aritméticas. El cálculo del resultado de la epresión $(1 \times 1) \times (0 \times 0)$ puede ser descripto por cualquiera de las siguientes transformaciones graduales:

$$\begin{array}{l} (1 \times 1) \times (0 \times 0) \;\to\; 1 \times (0 \times 0) \;\to\; 1 \times 0 \;\to\; 0 \\ (1 \times 1) \times (0 \times 0) \;\to\; (1 \times 1) \times 0 \;\to\; 1 \times 0 \;\to\; 0 \end{array} \tag{A.1}$$

Observamos que la simplificación procede por medio de una secuencia de **pasos de reescritura**. Cada paso tiene una dirección definida, de una expresión **origen** a otra **destino**; de aquí el uso de flechas, y no el de un símbolo de igualdad o similar, para denotar cada paso. Las expresiones **paso de reducción** y **secuencia de reducción** se usan comúnmente en la bibliografía para denotar los pasos de reescritura y las secuencias formadas por los mismos. Las secuencias de reducción también son conocidas como **reducciones** o *derivaciones*. En esta tesis notamos $t \twoheadrightarrow u$ si existe, al menos, una reducción con origen en el objeto t y destino en el objeto u. A la aplicación, o ejecución, de un paso, también se la conoce como **contracción**. Una **forma normal** es un objeto que no es origen de ningún paso. Por ejemplo, la expresión 0 es una forma normal para la simplificación de expresiones aritméticas. Cuando se utiliza la teoría de la reescritura para modelar una clase de cómputos, las formas normales suelen estar asociadas a los *resultados finales* de dichos cómputos. Un objeto t se dice **normalizante** si existe una forma normal u que verifica $t \twoheadrightarrow u$.

En la mayor parte de los casos, se utilizan **reglas de reescritura** para especificar las transformaciones válidas: cada paso debe corresponder a la aplicación de una regla. Un conjunto de reglas de reescritura forma la base de la definición de un **sistema de reescritura**. En nuestro ejemplo sobre simplificación de expresiones aritméticas, las reglas:

$$1 \times x \to x \qquad\qquad x \times 0 \to 0$$

alcanzan para justificar cada uno de los pasos en las secuencias detalladas en (A.1). Al aplicar una regla, cada *variable* que aparece en la misma puede ser reemplazada por cualquier expresión. P.ej., el paso $(1 \times 1) \times (0 \times 0) \to 1 \times (0 \times 0)$ corresponde a una aplicación de la regla $1 \times x \to x$, donde la variable x es reemplazada por la segunda ocurrencia de 1 desde la izquierda, en la expresión origen del paso. El reemplazo de una variable por una expresión más compleja al aplicar una regla de reescritura da lugar, p.ej., a la siguiente secuencia:

$$(1 \times 1) \times (0 \times 0) \to (1 \times 1) \times 0 \to 0$$

cuyo segundo paso corresponde a la regla $x \times 0 \to 0$, donde x se reemplaza por la expresión (1×1).

Un rápido repaso de algunos conceptos del **cálculo-lambda** permite introducir algunas nociones que se utilizan en esta tesis. El cálculo-lambda puede describirse como una formalización minimalista de la aplicación de una función a un argumento. La sintaxis básica provee únicamente un conjunto de *variables*, un constructor de *abstracción* que permite definir una función, y un segundo constructor para denotar la *aplicación* de una función a un argumento. P.ej. el término

$$(\lambda x.x + x + x)\, 3$$

denota la aplicación de la función $(\lambda x.x + x + x)$ al argumento 3. Las ocurrencias de la variable x en el subtérmino $x + x + x$ están **ligadas** por la abstracción λx. Los sistemas de reescritura de términos que, como el cálculo-lambda, incluyen mecanismos para ligar ocurrencias de variables, son conocidos como **sistemas de reescritura de términos de alto orden**. Los **sistemas de reescritura de términos de primer orden** son aquellos que no incluyen tales mecanismos.

El cálculo-lambda incluye una única regla de reescritura:

$$(\lambda x.s)u \longrightarrow \{x := u\}s$$

conocida como regla β. Aquí, $\{x := u\}s$ denota la **sustitución**, en el término s, de las ocurrencias (no ligadas) de x por u. Un ejemplo de paso de reducción es

$$(\lambda x.x + x + x)\, 3 \longrightarrow 3 + 3 + 3$$

Notar que este es un paso *atómico* en el modelo del cálculo-lambda: la aplicación de la sustitución $\{x := 3\}$ a $x + x + x$ se considera una operación externa al cálculo. Por otro lado, la sintaxis del cálculo-lambda no provee ningún mecanismo para *filtrar* los argumentos que puede aceptar una función.

El conjunto de transformaciones posibles en un sistema de reescritura puede describirse como un grafo, cuyos vértices son los objetos y cuyos ejes se corresponden con los pasos de reducción. Este grafo es conocido como el **espacio de reducciones**, o *espacio de derivaciones*, asociado al sistema de reescritura. Las secuencias de reducción se corresponden, exactamente, con los caminos del espacio de reducción. Los pares de objetos conectados forman la **relación de reducción**: el par $\langle t, u \rangle$ está en dicha relación si, y sólo si, $t \twoheadrightarrow u$, o sea, si existe una secuencia de reducción que tiene a t y a u como origen y destino respectivamente. Nótese que el espacio de reducción de un sistema de reescritura brinda un modelo más rico del mismo que su relación de reducción.

Los espacios de reducción suelen ser complejos, incluso los correspondientes a sistemas de reescritura sencillos. La Figura A.1 describe la fracción del sistema que modela la simplificación de expresiones aritméticas, formada por las distintas maneras de simplificar la expresión $(1 \times 1) \times (0 \times 0)$.

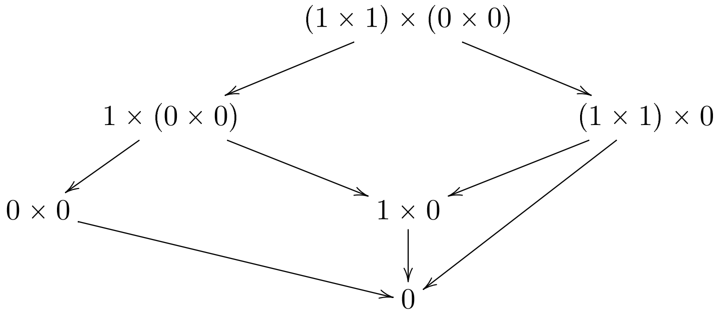


Figure A.1: Fracción de un espacio de reducción

Los conceptos y propiedades de sistemas de reescritura abordados en esta tesis están estrechamente relacionados con los espacios de reducciones.

La noción de **equivalencia entre reducciones** puede servir como guía para el estudio de espacios de reducciones. Dos reducciones se consideran equivalentes si comprenden, esencialmente, los mismos pasos de reducción, realizados en distinto orden. Es el caso de las reducciones $(1 \times 1) \times (0 \times 0) \to 1 \times (0 \times 0) \to 1 \times 0$ y $(1 \times 1) \times (0 \times 0) \to (1 \times 1) \times 0 \to 1 \times 0$, que se corresponden exactamente con el rombo superior en la Figura A.1. Destacamos que la coincidencia de origen y destino no es suficiente para que dos reducciones sean consideradas equivalentes. Por ejemplo, hay dos formas distintas, no equivalentes, de transformar $1 \times (1 \times 1)$ en 1×1, que consisten en dos aplicaciones distintas de la regla $1 \times x \to x$, donde se reemplaza la variable x, respectivamente, por (1×1) y por 1 (en el segundo caso, la regla aplica al subtérmino (1×1)).

Los llamados estudios de **estandarización** buscan definir subconjuntos minimales del conjunto de reducciones de un sistema de reescritura, que cubran completamente la relación de reducción. Una **clase de reducciones standard** debería incluir, al menos, una reducción de t a u, para cada par de objetos que verifiquen $t \twoheadrightarrow u$; este es el llamado criterio de **existencia** de reducciones standard. Idealmente, una clase de reducciones standard debería incluir *exactamente una* reducción de t a u por cada par que verifique $t \twoheadrightarrow u$; este es el llamado criterio de **unicidad** de reducciones standard.

Los trabajos sobre estandarización están ligados, en muchos casos, a la noción de

paso de reducción externo, caracterizándose como standard las reducciones en las que los pasos externos preceden a los internos. Según este criterio, en nuestro ejemplo sobre simplificaciones aritméticas, el paso $\underbrace{1 \times (2 \times 0)} \to 2 \times 0$ debe preceder, en una reducción standard, al paso $1 \times (\underbrace{2 \times 0}) \to 1 \times 0$; por lo tanto, la reducción $1 \times (2 \times 0) \to 2 \times 0 \to 0$ es standard, mientras que $1 \times (2 \times 0) \to 1 \times 0 \to 0$ no lo es.

Una **estrategia de reducción** puede pensarse como un "plan" para llevar a cabo una reducción partiendo de un determinado objeto. Formalmente, una estrategia puede definirse como una función, que dado un objeto t, indica un paso de reducción sobre t. Realizar este paso resulta en un nuevo objeto, el destino del paso elegido; llamemos a este objeto u. La estrategia se aplica a su vez sobre u, obteniéndose un nevo paso a aplicar, y así sucesivamente. Una estrategia también puede elegir, en lugar de un solo paso, un conjunto (no vacío) de pasos de reducción a aplicar sobre el objeto t. En tal caso, hablamos de **estrategias multipaso**. Los pasos elegidos sobre un objeto en una estrategia multipaso deben aplicarse *simultáneamente*, para lo cual puede apelarse a la noción de **desarrollo completo**[1] de un conjunto de pasos.

El objetivo, al definir estrategias de reducción, es obtener una forma normal a partir de cualquier término normalizante, mediante la *aplicación sistemática* de la estrategia. Formalmente, se dice que una estrategia de reducción es **normalizante** si, y sólo si, para todo objeto normalizante t, existe una secuencia $t_0, t_1, \ldots, t_n$, tal que $t = t_0$, t_n es una forma normal, y t_{i+i} es el objeto resultante de aplicar el, o los, paso/s de reducción indicados por la estrategia para t_i, para todo $i < n$. Se conoce como **normalización** al estudio de estrategias de reducción, incluyendo el desarrollo de técnicas para definir estrategias que resulten normalizantes, y de otras que permitan demostrar que una dada estrategia es normalizante.

El marco general descripto nos permite enunciar las contribuciones principales de esta tesis. Abordamos un estudio de características del espacio de reducción para distintos sistemas de reescritura de términos, según se detalla a continuación.

El Capítulo 3 es un estudio sobre *normalización*, enfocado particularmente en estrategias multipaso. Se presenta una demostración abstracta de normalización para estrategias multipaso, que da un conjunto de condiciones que resultan suficientes para garantizar que una estrategia es normalizante. Estas condiciones se refieren, algunas al sistema de reescritura para el que se define la estrategia, y otras a la estrategia en sí. También se define una estrategia multipaso para el *Pure Pattern Calculus* (`PPC`), y se demuestra que la estrategia definida es normalizante aplicando la demostración abstracta recién introducida. El `PPC` pertenece a la familia de los *cálculos con patrones*, que se focalizan en la formalización de la capacidad de *pattern matching* presente en los lenguajes de programación funcionales. Las características de este sistema de reescritura hacen que sea particularmente pertinente el uso de estrategias multipaso.

El Capítulo 4 es un estudio de *estandarización* para el *linear substitution calculus*. Definimos dos criterios distintos para considerar una reducción como standard en este sistema de reescritura; para ambos mostramos que cumplen el criterio de existencia, y para el segundo de ellos mostramos que también cumple con el criterio de unicidad, utilizando en este último caso una técnica de demostración novedosa. El *linear substitution*

[1] Usamos "desarrollo (completo)" como traducción al castellano de la locución inglesa "(complete) development".

calculus pertenece a la familia de los *cálculo con sustituciones explícitas*, cuyo foco es la formalización detallada de los distintos pasos que conlleva la *sustitución*, partiendo de un término t, de todas las ocurrencias de una variable x por otro término u. Destacamos que la bibliografía conocida por este autor incluye sólo un estudio de estandarización para un cálculo con sustituciones explícitas, a pesar de la proliferación de propuestas de definición de distintos sistemas de reescritura en esta familia, y de estudios concernientes a sus propiedades formales.

En el Capítulo 5, presentamos una caracterización de la *equivalencia de reducciones* para sistemas de reescritura de términos infinitaria de primer orden. La reescritura infinitaria de términos estudia los sistemas que admiten reducciones en las que intervienen términos infinitos, así como también reducciones que involucran una cantidad infinita de pasos, de las cuales puede determinarse un término destino, apelando a la noción de límite. La longitud de una reducción infinita puede, incluso, superar estrictamente el primer ordinal infinito, o sea ω: se admiten reducciones cuya longitud es $\omega + 1$, $\omega \times 2$, ω^2, etc..

A partir de esta propuesta para modelar la equivalencia de reducciones, presentamos una demostración alternativa del resultado de *compresión* de reducciones convergentes para sistemas de reescritura lineales a izquierda, en el que se establece que toda reducción convergente es equivalente a otra cuya longitud es, a lo sumo, el ordinal ω.

Finalizamos esta introducción destacando un rasgo común de las tres líneas de trabajo incluidas en esta tesis: se trata del uso de **modelos genéricos de reescritura**. Un modelo genérico brinda un marco para el estudio de propiedades de sistemas de reescritura, brindando definiciones abstractas de conceptos comunes tales como secuencia de reducción, espacio de reducciones, equivalencia de reducciones, etc.. A partir del marco que provee un modelo genérico, se pueden desarrollar *demostraciones abstractas* de propiedades, p.ej. vinculadas con la estandarización o la normalización. Los conceptos definidos y las demostraciones desarrolladas en un modelo genérico, resultan en consecuencia válidos para cualquier sistema de reescritura que pueda encuadrarse dentro del marco que provee dicho modelo.

El material incluido en los Capítulos 3 y 4 está basado en el modelo de los llamados *sistemas abstractos de reescritura*, *ARS* por sus siglas en inglés, utilizando la formulación desarrollada en [Mel96]. Por otra parte, el estudio sobre reescritura infinitaria del Capítulo 5 utiliza el modelo basado en la noción de *proof term*, tomando como punto de partida la formulación para reescritura finita de primer orden desarrollada en [BKdV03], Secciones 8.2 y 8.3. El Capítulo 2 de esta tesis es una descripción de las nociones fundamentales de estos dos modelos genéricos.

Destacamos que la presente tesis incluye contribuciones al desarrollo de los modelos genéricos utilizados, como ser la demostración abstracta de normalización incluida en el capítulo 3, desarrollada en el modelo de los sistemas abstractos de reescritura, y la extensión del modelo de proof terms para abarcar sistemas de reescritura infinitaria, que introducimos en el Capítulo 5.

Esta tesis puede ser considerada como un trabajo sobre el uso de modelos genéricos, para estudiar sistemas de reescritura cuyas características hacen particularmente desafiante el estudio de sus espacios de reducción.

A.2 Modelos genéricos de sistemas de reescritura

En este capítulo se introducen los dos modelos genéricos usados en esta tesis, a saber: el de los *Sistemas Abstractos de Reescritura*, o *ARS* por sus siglas en inglés, y el de los *proof terms*.

Un **Sistema Abstracto de Reescritura**, o **ARS**, es una estructura que modela a un sistema de reescritura. Los elementos básicos de un ARS son dos conjuntos, el de los **objetos** que se reescriben, notación $\mathcal{O}$, y el de los **pasos de reducción**, notación $\mathcal{R}$. La idea de “paso de reducción” en este modelo es similar al rol que tienen los pasos en un un espacio de reducción: un paso es un eje que liga un objeto **fuente** con uno **destino**. Esta idea se formaliza por medio de dos funciones: $\mathsf{src}, \mathsf{tgt} : \mathcal{R} \to \mathcal{O}$. Destacamos que en este modelo un objeto, así como un paso, son meramente elementos en un conjunto; no se incluye ninguna información *sintáctica*, acerca de la estructura de los términos, o del subtérmino correspondiente a un paso. Tampoco se incluye información sobre qué regla de reducción genera cada paso. Toda la información que se incluye en un ARS, por fuera de los conjuntos de objetos y pasos, y de las funciones que describen fuente y destino de cada paso, se produce por medio de *relaciones* definidas en el conjunto de pasos.

La principal de estas relaciones es la de **residuo**, notación $[\![\cdot]\!]$; es una relación ternaria. Se usa la notación $a[\![b]\!]a'$ para indicar $(a, b, a') \in [\![\cdot]\!]$. En tal caso, decimos que a' es un residuo de a después de b. La idea es que, siendo el origen de a el mismo que el de b (o sea, $\mathsf{src}(a) = \mathsf{src}(b)$), a' es un paso, cuyo origen es el *destino* de b (esto es, $\mathsf{src}(a') = \mathsf{tgt}(b)$), y que “proviene” de a. Dicho de otra forma, a' es (parte de) “lo que queda” de a en el objeto destino de b. Veamos un ejemplo en el sistema de simplificación de expresiones aritméticas; llamemos b al paso $t = (1 \times 1) \times (0 \times 0) \to 1 \times (0 \times 0) = u$. Notamos que el paso correspondiente al subtérmino 0×0 en u, proviene del correspondiente al mismo subtérmino en t. Si llamamos a y a' a los pasos correspondientes a 0×0 en t y u respectivamente, entonces tenemos $a[\![b]\!]a'$. Gráficamente:

$$
\begin{array}{ccc}
t = (1 \times 1) \times (0 \times 0) & \xrightarrow{\quad b \quad} & 1 \times (0 \times 0) = u \\
\downarrow a & a[\![b]\!]a' & \downarrow a' \\
(1 \times 1) \times 0 & & 1 \times 0
\end{array}
$$

En otros sistemas de reescritura, la relación de residuo es menos sencilla. Si consideramos los siguientes ejemplos en el cálculo-lambda:

$$
1)\quad \overbrace{(\lambda x.3)(\underbrace{(\lambda y.y)5}_{a})}^{b} \xrightarrow{b} 3
\qquad\qquad
2)\quad \overbrace{(\lambda x.xx)(\underbrace{(\lambda y.y)5}_{a})}^{b} \xrightarrow{b} (\underbrace{(\lambda y.y)5}_{a'})\ (\underbrace{(\lambda y.y)5}_{a''})
$$

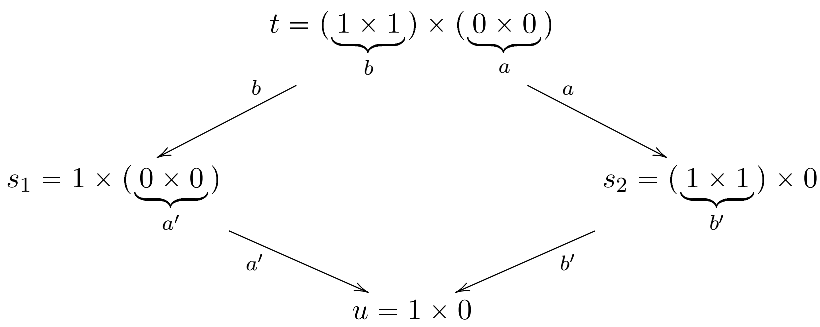
$$
3)\quad \overbrace{(\lambda x.\, \underbrace{(\lambda y.y)x}_{a})5}^{b} \xrightarrow{b} \underbrace{(\lambda y.y)5}_{a'}
$$

notamos que en 1), a no tiene ningún residuo después de b, mientras que en 2), $a$tiene dos residuos; en 3), el subtérmino correspondiente al paso a, que es $(\lambda y.y)x$, es “transformado” en $(\lambda y.y)5$, el subtérmino correspondiente al residuo de a después de b.

La mayor parte de los resultados que pueden obtenerse mediante el modelo ARS apelan también a la relación de **embedding** entre pasos. Es una relación binaria que notamos mediante el símbolo $<$, usado en forma infija. Un par $b < a$ en esta relación indica que b tiene, al menos potencialmente, la potestad de multiplicar, o bien de borrar, a; o sea, hacer que a tenga más de un residuo, o bien no tenga residuos, después de b. De acuerdo a esta idea, cualquier ARS que modele el cálculo-lambda debe incluir $b < a$ para los casos 1) y 2) de los ejemplos recién descriptos. Estos ejemplos sugieren una correlación entre la noción, semántica, de embedding, y el *anidamiento sintáctico* entre (los subtérminos correspondientes a los respectivos) pasos de reducción, en sistemas de reescritura de términos. En los modelos del cálculo-lambda como ARS descriptos en [Mel96], y también en la representación de otros sistemas de reescritura, en particular el *Pure Pattern Calculus* que estudiamos en el Capítulo 3, una condición necesaria para $b < a$ es que el paso b anide sintácticamente al paso a. En el Capítulo 4 mostramos una excepción a esta correlación entre embedding y anidamiento.

Destacamos que la visión de un sistema de reescritura que presenta el modelo ARS está orientada al *espacio de reducciones* del mismo.

Los elementos incluidos en un ARS, según lo descripto hasta el momento, permiten describir en forma abstracta varias nociones y propiedades relevantes de sistemas de reescritura. Dos pasos a y b cuyo origen coincide son **ortogonales** en este modelo, si aplicar a, y luego los residuos de) b después de a, produce el mismo efecto (esto es, tiene el mismo destino, y define la misma relación de residuos que) aplicar b, y luego los residuos de a después de b. El siguiente gráfico muestra un caso sencillo:



Aquí, a' es el único residuo de a después de b, y análogamente, b' es el único residuo de b después de a. En este ejemplo podemos notar, asimismo, que la secuencia $a; b'$ puede obtenerse **permutando** los dos pasos que forman la secuencia $b; a'$. A partir de la noción de permutación de pasos, se caracteriza la **equivalencia entre reducciones** en el modelo ARS: dos reducciones son equivalentes si, y sólo si, cada una de ellaas puede obtenerse como el resultado de una serie de permutaciones de pasos a partir de la otra. La relación de embedding da lugar a una noción de *paso externo*, a partir de la cual se deriva una caracterización abstracta de *reducción standard*.

El modelo ARS, incluyendo las relaciones de residuo y de embedding, tiene la riqueza suficiente para desarrollar *demostraciones abstractas* de propiedades relevantes. En [Mel96] se incluyen propiedades sobre estandarización y normalización, entre otros aspectos. En esta tesis se desarrollan una nueva demostración abstracta de normalización, y otra de estandarización, en este modelo.

En las demostraciones abstractas se establecen condiciones que debe verificar un sistema de reescritura para poder afirmar, para dicho sistema, la propiedad demostrada.

Estas condiciones se especifican en la forma de *axiomas*, cuyos enunciados se basan en las relaciones de residuo y de embedding. Estos axiomas permiten caracterizar, en forma abstracta, distintas características de un sistema de reeescritura, que resultan pertinentes para su estudio.

Finalmente, mencionamos que para el estudio de normalización en sistemas no-secuenciales desarrollado en el Capítulo 3 requiere del agregado, en la definición de un ARS, de una tercer relación entre pasos de reducción, llamada relación de *gripping*.

La noción de *proof term* es la base del otro modelo genérico de sistemas de reescritura usado en esta tesis. Como veremos, este modelo resulta menos abstracto que el basado en ARS. Existen varias formulaciones de este modelo, que apuntan a distintas familias de sistemas de reescritura; en esta tesis nos basamos en la que se presenta en [BKdV03] para reescritura de términos de primer order, cuyos conceptos principales presentamos a continuación.

Un **proof term** para un sistema de reescritura de términos T, es un término en una signatura que extiende la de T. Para cada regla se agrega un símbolo que denotará los pasos de reducción correspondientes a dicha regla. También se agrega un símbolo binario de concatenación, que se nota mediante el símbolo $\cdot$ usado en forma infija. Veamos algunos ejemplos de proof terms para el sistema de simplificaciones aritméticas, dándole a las reglas estos nombres: $\mu : 1 \times y \to y$, $\nu : y \times 0 \to 0$.

$$\begin{array}{rcl}
\mu(3) & : & \overbrace{1 \times \underline{3}} \;\to\; 3 \\
1 \times \nu(1 \times 1) & : & 1 \times (\overbrace{(\underline{1 \times 1}) \times 0}) \;\to\; 1 \times 0 \\
3 \times \nu(2 \times 1) \;\cdot\; \nu(3) & : & 3 \times (\overbrace{(\underline{2 \times 1}) \times 0}) \;\to\; \overbrace{\underline{3} \times 0} \;\to\; 0 \\
\mu(1) \times (2 \times 0) \;\cdot\; 1 \times \nu(2) \;\cdot\; \mu(0) & : & (\overbrace{1 \times \underline{1}}) \times (2 \times 0) \;\to\; 1 \times (\overbrace{\underline{2} \times 0}) \;\to\; \overbrace{1 \times \underline{0}} \;\to\; 0
\end{array}$$

En estos ejemplos, se indica el subtérmino correspondiente a cada paso con una llave, y el reemplazo de la variable y en la regla utilizada mediante subrayado. En el último caso, se aprovecha que la concatenación es asociativa para omitir un par de paréntesis.

Una característica destacable de este modelo es que los símbolos de regla, y también el de concatenación, pueden combinarse de distintas formas. Esto permite denotar la contracción *simultánea* y/o *localizada* de pasos de reducción, como se aprecia en los siguientes ejemplos:

$$\begin{array}{rcl}
\mu(1) \times \nu(2) & : & (1 \times 1) \times (2 \times 0) \;\multimap\!\!\!\to\; 1 \times 0 \\
\mu(\nu(2)) & : & 1 \times (2 \times 0) \;\multimap\!\!\!\to\; 0 \\
2 \times \big(\mu(1) \times 3 \;\cdot\; \mu(3)\big) & : & 2 \times ((1 \times 1) \times 3) \;\to\; 2 \times (1 \times 3) \;\to\; 2 \times 3
\end{array}$$

donde $\multimap\!\!\!\to$ denota la aplicación simultánea de pasos de reducción. Observamos que $\mu(1) \times \nu(2)$, $\mu(1) \times (2 \times 0) \;\cdot\; 1 \times \nu(2)$ y $(1 \times 1) \times \nu(2) \;\cdot\; \mu(1) \times 0$ son tres *proof terms distintos*, de forma tal que este modelo permite diferenciar la contracción simultánea de pasos, de su contrapartida secuencial. En esta tesis usamos la locución **actividad de contracción** para referirnos a las distintas formas de combinar pasos de reducción que pueden ser distinguidas en el modelo de *proof terms*.

La **equivalencia de reducciones** puede describirse en el modelo de *proof terms* a partir de la noción de permutación de pasos. Se define un conjunto de ecuaciones que formaliza una permutación de pasos; dos reducciones (o más generalmente, dos *actividades de contracción*) se consideran equivalentes si, y sólo si, la equivalencia entre los *proof terms* que las representan puede concluirse mediante lógica ecuacional a partir de dichas ecuaciones. Las ecuaciones describen que tanto la sucesión formada por un paso a seguido de (los residuos de) b, como la formada de b seguido por (los residuos de) a, son equivalentes a la contracción simultánea de los dos pasos. La equivalencia entre las dos secuencias se establece mediante la equivalencia de cada una de ellas con la versión simultánea. P.ej., se establece que: $(1 \times 1) \times \nu(2) \cdot \mu(1) \times 0 \approx \mu(1) \times \nu(2) \approx \mu(1) \times (2 \times 0) \cdot 1 \times \nu(2)$, formalizándose de esta forma la permutación entre los dos pasos de la secuencia $(1 \times 1) \times (2 \times 0) \to (1 \times 1) \times 0 \to 1 \times 0$.

A partir de esta caracterización de la equivalencia de reducciones, en [BKdV03] se obtienen resultados de *estandarización* para sistemas de reescritura de términos de primer orden.

Finalmente, mencionamos que una segunda caracterización de la equivalencia entre reducciones usando *proof terms*, basada en la noción de *proyecciones*, también aparece en [BKdV03], en donde se establece la equivalencia entre estas dos caracterizaciones, y también con otras que también se describen allí. Se introduce aquí la caracterización de la equivalencia mediante permutaciones sucesivas, porque es la que se extiende en el Capítulo 5 de esta tesis a sistemas de reescritura infinitarios.

A.3 Normalización

La noción de *paso necesario* está estrechamente relacionada con el estudio de la normalización de sistemas de reescritura, que es el tema general de este capítulo. Un paso con origen en un objeto t se dice **necesario** si su contracción resulta ineludible para obtener una forma normal a partir de t; o sea, si cualquier reducción con origen es t y cuyo destino es una forma normal, incluye a ese paso, o bien a al menos uno de sus residuos. Varios estudios de normalización están basados en la noción de paso necesario. En particular, en [HL91] se demuestra que la contraccón sistemática de pasos necesarios es normalizante.

Por otro lado, los enfoques basados en la noción de paso necesario no son aplicables en sistemas de reescritura que admiten términos, que no son formas normales, y para los cuales ninguno de sus pasos resulta necesario. Un ejemplo profusamente mencionado en la literatura es el llamado **disyunción paralela**, que incluye las siguientes reglas:

$$\mathtt{or}(x, \mathtt{tt}) \to \mathtt{tt} \qquad\qquad \mathtt{or}(\mathtt{tt}, x) \to \mathtt{tt}$$

El término $\mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{or}(\mathtt{ff},\mathtt{tt}))$ incluye dos pasos, correspondientes a los subtérminos $\mathtt{or}(\mathtt{tt},\mathtt{ff})$ y $\mathtt{or}(\mathtt{ff},\mathtt{tt})$. Las siguientes secuencias de reducción

$$\begin{array}{lclcl} \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{or}(\mathtt{ff},\mathtt{tt})) & \to & \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{tt}) & \to & \mathtt{tt} \\ \mathtt{or}(\mathtt{or}(\mathtt{tt},\mathtt{ff}),\mathtt{or}(\mathtt{ff},\mathtt{tt})) & \to & \mathtt{or}(\mathtt{tt},\mathtt{or}(\mathtt{ff},\mathtt{tt})) & \to & \mathtt{tt} \end{array}$$

cuyo destino es una forma normal, muestran que ninguno de los dos pasos del término origen son necesarios: el paso de la izquierda (respect., de la derecha) no es utilizado en la primer (respect., en la segunda) secuencia. Incluir términos, que no son formas normales, y que no incluyen ningún paso necesario, es condición suficiente para considerar a un sistema de reescritura como **no secuencial**; la definición precisa de esta noción escapa al presente resumen.

Varios trabajos coinciden en señalar la pertiencia de considerar estrategias multipaso para obtener estrategias normalizantes en sistemas no-secuenciales. En particular, en [SR93] se demuestra que la contracción sistemática de *conjuntos necesarios de pasos* es normalizante para sistemas de reescritura de términos no-secuenciales de primer orden, como es el caso de la disyunción paralela. La noción de **conjunto necesario de pasos** generaliza la de paso necesario: un conjunto de pasos $\mathcal{A}$ con origen en un objeto t es necesario si cualquier reducción con origen en t cuyo destino es una forma normal incluye al menos un paso en $\mathcal{A}$, o alguno de sus residuos.

En esta tesis se utiliza el modelo de los ARS para desarrollar una **demostración abstracta de normalización**. Esta demostración se basa en la estructura de la que aparece en [SR93], utilizando también algunas ideas que se proponen en [vO99]. Nuestra prueba extiende la de [SR93], dado que el modelo ARS puede aplicarse a sistemas de reescritura de alto orden. Por otro lado, se debe requerir una condición adicional sobre los conjuntos de pasos: además de ser conjuntos necesarios, deben ser *non-gripping*, condición surgida en el estudio abstacto de desarrollos completos incluido en [Mel96].

En la introducción se mencionó que la sintaxis del cálculo-lambda no incluye mecanismos para *filtrar* los posibles argumentos de una función; una abstracción de la forma $\lambda x.s$ puede ser aplicada a *cualquier* término. Esta situación contrasta con la práctica habitual de los lenguajes de programación funcionales. Estos lenguajes incluyen una característica conocida como **pattern matching**, por la que al definirse una función, pueden especificarse restricciones sobre la forma de sus argumentos. Tomemos esta definición en Haskell:

```
length []       = 0
length (x:xs)   = 1 + length xs
```

La función `length` así definida sólo puede ser aplicada a listas; si se aplica esta función a, p.ej., un número, se produce un **error de matching**. Además, presenta dos definiciones distintas, para listas vacías (notación `[]`) y no vacías (notación `x:xs`) respectivamente.

Los **cálculos con patrones** tienen como objetivo modelar formalmente el fenómeno del *pattern matching*. El Pure Pattern Calculus, o PPC, pertenece a esta familia de sistemas de reescritura. Varios cálculos con patrones, entre ellos el PPC, incluyen un constructor de abstracción generalizada, de la forma $\lambda p.s$, donde p es un *patrón*. Dichos cálculos incluyen una generalización de la regla β, de la forma

$$(\lambda p.s)u \;\to\; \{p/u\}s$$

donde $\{p/u\}s$ es el resultado del *matching* del argumento u respecto del patrón p. La definición del matching de argumento contra patrón es uno de los aspectos principales en la definición de un cálculo con patrones.

En estos cálculos con patrones, si $\mathtt{p}$ es un constructor de datos, entonces el siguiente es un paso de reducción válido

$$(\lambda \mathtt{p}\, x\, y.\, y)\ (\mathtt{p}\ 3\, 4) \;\to\; 4$$

mientras que la aplicación $(\lambda \mathtt{p}\, x\, y.\, y)\ 3$ desencadena un **mecanismo de error** especificado en la definición del cálculo, dado que el matching del argumento 3 respecto del patrón $\mathtt{p}\, x\, y$ es imposible. En el PPC, los errores de matching reducen en un paso a un término que representa a la función identidad, que notaremos como I. Por lo tanto, en PPC tenemos

$$(\lambda \mathtt{p}\, x\, y.\, y)\ 3\ \rightarrow\ I$$

Esta defnición del PPC permite modelar las alternativas en la definición de una función; cfr. [JK09].

La sintaxis del PPC se describe en la Sección 3.4.1, y en [JK09]; los ejemplos que brindamos a continuación usan una versión simplificada de dicha sintaxis. Una característica saliente del PPC es que *cualquier* término puede ser un patrón. En particular, un patrón puede incluir ocurrencias libres de variables, para lo cual se indica explícitamente, para cada abstracción, cuáles son las variables que liga. P.ej. la función identidad puede definirse en PPC mediante el término $\lambda_{\{x\}}x.x$.

En el término

$$t = (\lambda_{\{x\}}x.(\lambda_{\{y,z\}}x(yz).y))$$

las dos ocurrencias de x están ligadas por el abstractor exterior. Esto permite generar un patrón concreto a partir de la especificación genérica $\lambda_{\{y,z\}}x(yz)$, aplicando t a un término adecuado. En esta reducción

$$(\lambda_{\{x\}}x.(\lambda_{\{y,z\}}x(yz).y))\,\mathtt{a}\,(\mathtt{a}(34)) \rightarrow (\lambda_{\{y,z\}}\mathtt{a}(yz).y)\,(\mathtt{a}(34)) \rightarrow 3$$

se aplica t al constructor de datos a; como consecuencia, se obtiene una función que sólo acepta, como argumentos, estructuras de datos sobre ese constructor. Si aplicamos t a una *función*, el patrón concreto será el resultado de un cómputo que se lleva a cabo *dentro del patrón*, como en este caso:

$$\begin{aligned}
&(\lambda_{\{x\}}x.(\lambda_{\{y,z\}}x(yz).y))(\lambda_{\{x',y'\}}x'y'.\mathtt{p}y'x')(\mathtt{p}34)\\
&\quad\rightarrow (\lambda_{\{y,z\}}(\lambda_{\{x',y'\}}x'y'.\mathtt{p}y'x')(yz).y)\,(\mathtt{p}34)\\
&\quad\rightarrow (\lambda_{\{y,z\}}\mathtt{p}zy.y)\,(\mathtt{p}34) \rightarrow 4
\end{aligned}$$

Estos ejemplos muestran la naturaleza *dinámica* de los patrones en PPC, que permiten modelar formas de *polimorfismo* no presentes en los lenguajes funcionales utilizados actualmente en el ámbito del desarrollo de software; cfr. [JK06b, JK09] al respecto.

Destacamos que la defnición del matching de PPC evita los problemas respecto de la estabilidad del cálculo que podrían provenir de aceptar patrones como, p.ej., $x\,y$. P.ej. en el término $(\lambda_{\{x,y\}}xy.x)((\lambda_{\{z\}}z.z)3)$, el matching del argumento $(\lambda_{\{z\}}z.z)3$ respecto del patrón $\lambda_{\{x,y\}}xy$ no es exitoso y tampoco produce un error de matching; el único paso de reducción con origen en este término es el que corresponde al subtérmino $(\lambda_{\{z\}}z.z)3$.

Consideremos ahora la estructura de datos e ⟨nombre⟩ ⟨género⟩ ⟨facultad⟩, que representa a un estudiante. La función $\lambda_{\{x\}}\mathtt{e}\,x\,\mathtt{v}\,\mathtt{i}\ .\ x$ permite recuperar el nombre de un estudiante varón que estudia ingeniería. El término

$$(\lambda_{\{x\}}\mathtt{e}\,x\,\mathtt{v}\,\mathtt{i}\ .\ x)(\mathtt{e}(I\mathtt{a})(I\mathtt{m})(I\mathtt{d}))$$

incluye tres pasos de reducción, correspondientes a $(I\mathtt{a})$, $(I\mathtt{m})$ y $(I\mathtt{d})$ respectivamente. Las secuencias de reducción

$$\begin{array}{lllll}
(\lambda_{\{x\}}\mathtt{e}\,x\,\mathtt{v}\,\mathtt{i}\ .\ x)(\mathtt{e}(I\mathtt{a})(I\mathtt{m})(I\mathtt{d})) & \rightarrow & (\lambda_{\{x\}}\mathtt{e}\,x\,\mathtt{v}\,\mathtt{i}\ .\ x)(\mathtt{e}(I\mathtt{a})\mathtt{m}(I\mathtt{d})) & \rightarrow & I\\
(\lambda_{\{x\}}\mathtt{e}\,x\,\mathtt{v}\,\mathtt{i}\ .\ x)(\mathtt{e}(I\mathtt{a})(I\mathtt{m})(I\mathtt{d})) & \rightarrow & (\lambda_{\{x\}}\mathtt{e}\,x\,\mathtt{v}\,\mathtt{i}\ .\ x)(\mathtt{e}(I\mathtt{a})(I\mathtt{m})\mathtt{d}) & \rightarrow & I
\end{array}$$

muestran que ninguno de estos pasos es necesario, y que por lo tanto, el PPC es un sistema de reescritura *no-secuencial*. Por otro lado, destacamos que $\{(I\mathtt{m}), (I\mathtt{d})\}$ es un *conjunto necesario de pasos*. El segundo paso de la primer reducción puede explicarse como sigue: los segundos argumentos de la estructura encabezada por e en arguemto y patrón son dos constantes distintas, m y v respectivamente. Esta falla en un argumento alcanza para disparar el mecanismo de error de matching. En la segunda reducción, un argumento análogo aplica al tercer argumento.

En este capítulo definimos una **estrategia multipaso para el** PPC, y demostramos que esta estrategia es normalizante, apelando a la demostración abstracta mencionada anteriormente. Para esto mostramos que el PPC puede modelarse adecuadamente como un ARS, verificándose todas las condiciones impuestas por la demostración abstracta. La estrategia definida elige conjuntos necesarios y *non-gripping* de pasos. Destacamos que esta estrategia se comporta exactamente como la *leftmost-outermost* para los términos del PPC que tienen correspondencia inmediata en el cálculo-lambda, resultando "monopaso" para dichos términos.

A.4 Estandarización para el *linear substitution calculus*

Como se indica en la introducción, la sustitución es considerada como una operación externa en el cálculo-lambda. Por ejemplo, el siguiente:

$$(\lambda x.x + x + x) \;\rightarrow\; 3 + 3 + 3$$

es un paso de reducción atómico. Por otro lado, el cómputo de sustituciones es un elemento relevante en la *evaluación de programas en los lenguajes funcionales*, lo que genera interés por modelos formales detallados de esta operación. El objetivo de los **cálculos con sustituciones explícitas**, o **cálculos ES**, es modelar detalladamente la aplicación de una sustitución a un término. Brindamos una pequeña descripción de algunas facetas de los cálculos ES, tomando como ejemplo una variación del cálculo $\lambda\mathtt{x}$, [Ros92, BR95], en la que incluimos constantes.

Los cálculos ES proveen una construcción para denotar, explícitamente, la aplicación de una sustitución a un término. Así, si s y u son términos, entonces $s[x/u]$ es un término bien formado. Se incluye una regla de reescritura análoga a la regla β del cálculo-lambda, de la forma:

$$(\lambda x.s)u \;\rightarrow\; s[x/u]$$

A diferencia de la regla β, la sustitución $[x/u]$ sólo se *genera*, no se *evalúa*. Se definen reglas de reescritura adicionales para modelar detalladamente la evaluación de una sustitución a un término. En el cálculo que usamos como ejemplo, estas reglas son:

$$\begin{array}{lll} (t_1 t_2)[x/u] \to (t_1[x/u])(t_2[x/u]) & & (\lambda y.t)[x/u] \to \lambda y.t[x/u] \\ x[x/u] \to u & \mathtt{c}[x/u] \;\rightarrow\; \mathtt{c} & y[x/u] \to y \qquad \text{if } y \neq x \end{array}$$

Una sustitución se propaga (reglas en la primer línea), generándose copias; cada copia o bien se aplica, o bien se elimina (reglas en la segunda línea).

La propagación de sustituciones implica que los espacios de reducción de los cálculos ES tienden a ser extremadamente complejos. P.ej. para simular el paso de reducción del cálculo-lambda $(\lambda x.\mathtt{p}\,x(\mathtt{s}\,x))\,3 \to p\,3(s\,3)$ hacen falta ocho pasos de reducción en $\lambda\mathtt{x}$, que pueden ordenarse en formas distintas, dando lugar a una gran diversidad de distintas reducciones. Una de estas reducciones es:

$$\begin{array}{llllll}
(\lambda x.\mathtt{p}x(\mathtt{s}x))\,3 & \to & (\mathtt{p}x(\mathtt{s}x))\,[x/3] & \to & (\mathtt{p}x)\,[x/3]\,((\mathtt{s}x)\,[x/3]) \\
 & \to & \mathtt{p}\,[x/3]\,x\,[x/3]\,((\mathtt{s}x)\,[x/3]) & \to & \mathtt{p}\,[x/3]\,x\,[x/3]\,(\mathtt{s}\,[x/3]\,x\,[x/3]) \\
 & \to & \mathtt{p}\,x\,[x/3]\,(\mathtt{s}\,[x/3]\,x\,[x/3]) & \to & \mathtt{p}\,3\,(\mathtt{s}\,[x/3]\,x\,[x/3]) \\
 & \to & \mathtt{p}\,3\,(\mathtt{s}\,x\,[x/3]) & \to & \mathtt{p}\,3\,(\mathtt{s}\,3)
\end{array}$$

Los **cálculos ES a distancia** han sido propuestos recientemente. Estos cálculos ES evitan la propagación y copia de las sustituciones, permitiendo que una sustitución se aplique a una ocurrencia *distante* de la variable involucrada. Estos cálculos incluyen una regla de la forma

$$C[\![x]\!]\,[x/u] \;\to\; C[\![u]\!]\,[x/u] \tag{A.2}$$

donde C es un contexto arbitrario que no liga la ocurrencia de x en $C[\![x]\!]$. De esta forma, se obtienen cálculos cuyos espacios de reducciones tienen un menor grado de complejidad. Este capítulo estudia el **linear substitution calculus**, o $\lambda_{\mathtt{lsub}}^{\sim}$, un cálculo ES a distancia, que agrega la siguiente regla para eliminar sustituciones superfluas, de acuerdo a la idea de *garbage collection*:

$$t\,[x/u] \;\to\; t \qquad \text{if } x \notin \mathtt{fv}(t) \tag{A.3}$$

El paso de reducción $(\lambda x.\mathtt{p}\,x(\mathtt{s}\,x))\,3 \to p\,3(s\,3)$ puede emularse en $\lambda_{\mathtt{lsub}}^{\sim}$ como sigue:

$$\begin{array}{llllll}
(\lambda x.\mathtt{p}x(\mathtt{s}x))\,3 & \to & (\mathtt{p}x(\mathtt{s}x))\,[x/3] & \to & (\mathtt{p}\,3\,(\mathtt{s}x))\,[x/3] \\
 & \to & (\mathtt{p}\,3\,(\mathtt{s}\,3))\,[x/3] & \to & (\mathtt{p}\,3\,(\mathtt{s}\,3))
\end{array}$$

Esta reducción resulta más sencilla que la correspondiente a un cálculo ES con propagación, como la desarrollada anteriormente para $\lambda\mathtt{x}$.

En este capítulo se realiza un estudio de *estandarización* para el cálculo $\lambda_{\mathtt{lsub}}^{\sim}$, basado en el modelo ARS. Dos características de este sistema de reescritura hacen que modelarlo como un ARS resulte una tarea no trivial. Una de ellas es la existencia de distintos pasos de reducción con origen en el mismo término, que corresponden exactamente al mismo subtérmino. Es el caso de dos ocurrencias de la misma variable, que se corresponden con dos pasos distintos generados por la regla (A.2). P.ej. el subtérmino correspondiente a los dos pasos en el término $(xx\,[y/z])y$ es $xx\,[y/z]$. Esto provoca que para identificar un paso de reducción, no alcanza con el subtérmino correspondiente. Para pasos generados por la regla (A.2), el contexto C debe tenerse en cuenta. El otro aspecto problemático de $\lambda_{\mathtt{lsub}}^{\sim}$ es que la relación de *embedding* no se corresponde con el anidamiento sintáctico de pasos de reducción. Recordemos que si a y b son pasos con origen en el mismo término, entonces $a < b$ indica que la contracción de a podría duplicar, o bien eliminar, b. Si consideramos el término $t = x\,[x/y]\,[y/z]$ y llamamos a_x y a_y a los pasos generados por la regla (A.2) correspondientes a las ocurrencias de x e y respectivamente, notamos que a_y anida sintácticamente a a_x, como se aprecia en la siguiente figura:

$$\overbrace{\underbrace{x\,[x/y]}_{a_x}\,[y/z]}^{a_y}$$

Por otro lado, la contracción de estos pasos resulta en:

$$x[x/y][y/z] \xrightarrow{a_x} y[x/y][y/z] = t_1 \qquad\qquad x[x/y][y/z] \xrightarrow{a_y} x[x/z][y/z] = t_2$$

Se observa que la contracción de a_x duplica a_y; nótense las dos ocurrencias de y en t_1. Por lo tanto, en cualquier modelo de $\lambda^{\sim}_{\mathtt{lsub}}$ como un ARS, la relación de *embedding* debe incluir el par $a_x < a_y$, lo que se contradice con el anidamiento sintáctico.

Un primer ARS que describe $\lambda^{\sim}_{\mathtt{lsub}}$, definido teniendo en cuenta las peculiaridades recién descriptas, permite obtener resultados de existencia y unicidad de reducciones standard para este cálculo, apelando a demostraciones abstractas desarrolladas en [Mel96].

Sin embargo, estos resultados no resultan satisfactorios, debido a la estrecha relación que existe entre $\lambda^{\sim}_{\mathtt{lsub}}$ y las *proof nets* de la lógica lineal. Se puede establecer una equivalencia operacional fuerte entre términos de $\lambda^{\sim}_{\mathtt{lsub}}$ y *proof nets*, que se convierte en un isomorfismo si se considera, en lugar del conjunto de términos de $\lambda^{\sim}_{\mathtt{lsub}}$, su cociente por la relación de equivalencia generada por tres ecuaciones. Estas ecuaciones reflejan que la ubicación precisa de una sustitución dentro de un término es, en muchos casos, irrelevante; una de ellas es:

$$t[x/u][y/s] \sim t[y/s][x/u] \qquad\qquad \text{if } x \notin \mathtt{fv}(s) \ \wedge \ y \notin \mathtt{fv}(u)$$

En este isomorfismo, una *proof net* se corresponde, exactamente, con una clase de equivalencia de términos. Por lo tanto, los resultados que se obtengan a partir de un modelo de $\lambda^{\sim}_{\mathtt{lsub}}$ como ARS, cuyos objetos sean las *clases de equivalencia* del conjunto cociente recién mencionado, serán válidos también para *proof nets*.

Para obtener un ARS con estas características, establecemos una biyección entre los conjuntos de pasos de reducción de términos equivalentes, valiéndonos de la técnica de *etiquetado* (*labeling* en inglés), mostrando que la relación de residuos es una bisimulación respecto de esta biyección. Como la relación de *embedding* del primer ARS definido no es invariante respecto de la biyección entre pasos de reducción, definimos una nueva relación de *embedding*, que es una restricción de la anterior, y que sí resulta invariante.

De esta forma obtenemos un segundo ARS, que modela $\lambda^{\sim}_{\mathtt{lsub}}$ considerando que el conjunto de objetos que se reescriben es el cociente definido en el conjunto de términos, resultando cada término individual un mero representante de su clase de equivalencia. Este ARS cumple con las condiciones exigidas para el resultado de *existencia* de reducciones standard enunciado en [Mel96], pero no con aquellas requeridas para el resultado correspondiente de *unicidad*. A pesar de esto, obtenemos un resultado de unicidad de reducciones standard para $\lambda^{\sim}_{\mathtt{lsub}}$ considerado como un sistema de reescritura de clases de equivalencia de términos, por medio del desarrollo de una nueva *demostración abstracta de estandarización* en el modelo ARS.

A.5 Equivalencia de reducciones para reescritura infinitaria

Consideremos los sistemas de reescritura de términos T_1 y T_2, que definimos a continuación. El sistema T_1 incluye al número 1, el símbolo de la suma, un símbolo unario l, el constructor de listas que denotamos mediante el símbolo :, "dos puntos", y la regla de reescritura

$$l(x) \to x : l(x+1)$$

El sistema T_2 incluye las constantes a y b, y las reglas

$$a \rightarrow b \qquad\qquad b \rightarrow a$$

Podemos construir reducciones infinitas tanto en T_1 como en T_2. Para T_1, consideremos

$$l(1) \rightarrow 1 : l(2) \rightarrow 1 : 2 : l(3) \rightarrow \ldots$$

donde $2, 3, \ldots$ abrevian $1+1, (1+1)+1$, etc.. Para T_2, tenemos esta secuencia

$$a \rightarrow b \rightarrow a \rightarrow b \rightarrow a \rightarrow \ldots$$

Podemos apreciar una diferencia importante entre estas dos reduccciones, observando las secuencias de términos de destino parciales de cada una, que son respectivamente:

$$\langle 1 : l(2), 1 : 2 : l(3), 1 : 2 : 3 : l(4) \ldots \rangle \qquad \text{y} \qquad \langle b, a, b, a \ldots \rangle$$

No es difícil aprehender que la secuencia de la izquierda *converge*, con límite en el *término infinito* $1 : 2 : 3 : 4 : \ldots$, mientras que la secuencia de la derecha *diverge*. Notamos que los dos sistemas de reescritura presentados pueden describirse fácilmente en lenguajes de programación funcionales; consideremos p.ej. estas definiciones en Haskell:

```
natlist n = n : natlist (n+1)
diva      = divb
divb      = diva
```

Observamos que mientras la evaluación de `natlist 1` genera la lista `[1,2,3,4 ...]`, la de la expresión `diva` continúa indefinidamente, sin entregar ningún resultado parcial.

Estas consideraciones motivan el estudio de **sistemas de reescritura de términos infinitarios**. La noción de **convergencia** es particularmente relevante en este ámbito. Varios criterios de convergencia han sido propuestos en la literatura. En esta tesis, adoptamos el criterio de **convergencia fuerte**, según el cual una secuencia de reducción infinita[2] es convergente si, y sólo si, la secuencia formada por la *profundidad* de cada paso tiende a infinito. P.ej. la secuencia $l(1) \rightarrow 1 : l(2) \rightarrow 1 : 2 : l(3) \rightarrow \ldots$ es fuertemente convergente, pues el i-ésimo paso se realiza a profundidad $i-1$, más precisamente, debajo de $i-1$ ocurrencias anidadas del constructor de lista. Para las secuencias convergentes infinitas[3] , se puede establecer como *término destino* el límite de los destinos de los prefijos de dicha secuencia. Notamos $t \twoheadrightarrow u$ para indicar la existencia de una reducción, ya sea finita o infinita, con origen en t y destino en u.

En los sistemas de reescritura infinitarios, la infinitud se manifiesta de diversas formas. Veamos algunos ejemplos utilizando las reglas de reescritura $f(x) \rightarrow g(x)$ y $g(x) \rightarrow k(x)$. En un sistema de reescritura infinitario, cada paso de reducción puede involucrar *términos infinitos*. Si abreviamos como f^ω el término infinito $f(f(f \ldots))$, que consta de una cantidad infinita de ocurrencias, todas del símbolo f, entonces el siguiente es un paso de reducción válido: $f^\omega = f(f^\omega) \rightarrow g(f^\omega)$. Además, las *secuencias de reducción* pueden tener una longitud infinita. Un ejemplo es $f^\omega \rightarrow g(f^\omega) \rightarrow g(g(f^\omega)) \rightarrow \ldots \rightarrow g(\ldots g(f(f^\omega)) \ldots) \rightarrow g(\ldots g(g(f^\omega)) \ldots) \rightarrow \ldots$ Esta secuencia es convergente, siendo g^ω su término destino. En lo sucesivo, denotaremos una secuencia de

[2] más precisamente, una secuencia cuya longitud es un *ordinal límite*.

[3] otra vez, nos referimos a secuencias cuya longitud es un ordinal límite.

esta forma como $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega$. A partir de este último término, la secuencia puede continuar, p.ej. con el paso $g^\omega \to k(g^\omega)$, generándose una secuencia cuya longitud es $\omega + 1$, donde ω es el primer ordinal infinito. A su vez, la secuencia $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega \to k(g^\omega) \to k(k(g^\omega)) \twoheadrightarrow k^\omega$ es de longitud $\omega \times 2$. Puede demostrarse la existencia, para cualquier ordinal numerable α, de una reducción fuertemente convergente cuya longitud es exactamente α.

Este capítulo es un estudio de la **equivalencia de reducciones** en sistemas de reescritura de primer orden infinitarios, basado en el modelo de *proof terms*. Se toma como punto de partida la definición de *proof terms* y la caracterización de la equivalencia entre reducciones por permutaciones sucesivas utilizando *proof terms*, que aparecen en [BKdV03], Secciones 8.2 y 8.3. Para ello, se extiende a la reescritura infinitaria la noción de *proof term*, permitiendo denotar reducciones infinitas y/o sobre términos infinitos, mediante *proof terms*. Se pone especial cuidado en la denotación de la concatenación de una secuencia infinita de reducciones. También se extiende la caracterización de la *equivalencia entre reducciones por permutaciones sucesivas*, utilizando *lógica ecuacional* sobre *proof terms*: dos reducciones resultan equivalentes si, y sólo si, la ecuación $\psi \approx \phi$ puede concluirse, utilizando lógica ecuacional, a partir de un conjunto básico de ecuaciones, donde ψ y ϕ son *proof terms* que representan las dos reducciones en cuestión. Destacamos que esta forma de caracterizar la equivalencia de reducciones infinitarias es una aplicación de lógica ecuacional en un contexto infinitario.

Para obtener la caracterización mencionada de la equivalencia entre reducciones infinitarias se agrega, a las reglas que definen la clausura por equivalencia y por operaciones, una regla que permite apelar al concepto de *límite* en un razonamiento ecuacional: dos *proof terms* se consideran equivalentes, si son, cada uno de ellos, el límite de una secuencia de *proof terms*, tal que las distancias entre los elementos sucesivos de las dos secuencias tiende a cero. El concepto de *profundidad mínima* es utilizado para definir la distancia entre dos *proof terms*: separando la actividad que denotan dos proof terms en una parte común y otra que refleja la diferencia entre ellos, la distancia entre los proof terms es inversamente proporcional a la profundidad mínima de la parte en que difieren.

Destacamos que la caracterización obtenida modela adecuadamente casos en los cuales debe permutarse un paso respecto de una cantidad *infinita* de pasos, y/o realizar una *cantidad infinita de permutaciones*, para obtener la equivalencia entre dos reducciones. Damos algunos ejemplos, utilizando las reglas $f(x) \to g(x)$, $g(x) \to k(x)$ y $m(x) \to n(x)$. Para transformar la secuencia

$$m(f^\omega) \to m(g(f^\omega)) \to m(g(g(f^\omega))) \twoheadrightarrow m(g^\omega) \to n(g^\omega)$$

en la equivalente

$$m(f^\omega) \to n(f^\omega) \to n(g(f^\omega)) \to n(g(g(f^\omega))) \twoheadrightarrow n(g^\omega)$$

el último paso de la primer reducción debe permutarse con una cantidad *infinita* de pasos, pues se corresponde con el primer paso de la segunda reducción. Para transformar

$$f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega \to k(g^\omega) \to k(k(g^\omega)) \twoheadrightarrow k^\omega$$

en la secuencia equivalente

$$f^\omega \to g(f^\omega) \to k(f^\omega) \to k(g(f^\omega)) \to k(k(f^\omega)) \twoheadrightarrow k^\omega$$

cada uno de los infinitos pasos correspondientes a la regla $g(x) \to k(x)$ debe permutarse con infinitos pasos correspondientes a $f(x) \to g(x)$.

Demostramos que la representación de reducciones mediante *proof terms* es **completa**: para cada secuencia de reducción cuya longitud es un ordinal numerable, existe un *proof term* que la denota. Adicionalmente, mostramos que este *proof term* es *único*, salvo por la asociatividad del símbolo que representa la concatenación. Para demostrar esta afirmación de unicidad, se extiende la noción de "expresiones iguales salvo asociatividad de un operador binario" al términos infinitarios.

Finalmente, utilizamos la formalización de la equivalencia entre reducciones mediante *proof terms*, para desarrollar una demostración alternativa del resultado de **compresión** de reducciones infinitarias. Concretamente, demostramos que cualquier reducción[4] es equivalente a otra, cuya longitud es a lo sumo ω. P.ej., ya mencionamos que la secuencia $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega \to k(g^\omega) \to k(k(g^\omega)) \twoheadrightarrow k^\omega$, cuya longitud es $\omega \times 2$, es equivalente a $f^\omega \to g(f^\omega) \to k(f^\omega) \to k(g(f^\omega)) \to k(k(f^\omega)) \twoheadrightarrow k^\omega$, secuencia de longitud ω. Destacamos que el resultado demostrado representa una versión del resultado de compresión más fuerte que las que aparecen en la literatura conocida por el autor.

A.6 Conclusiones

Reseñamos las principales contribuciones realizadas en esta tesis.

Respecto de la **normalización**, presentamos una nueva demostración abstracta en el modelo ARS, que puede utilizarse para estrategias multipaso, y para sistemas de reescritura de términos tanto de primer orden como de alto orden. Definimos una estrategia multipaso para el *Pure Pattern Calculus*, un cálculo con patrones no-secuencial, y demostramos que dicha estrategia es normalizante, por medio de la demostración abstracta mencionada más arriba.

Respecto de la **estandarización**, obtenemos varios resultados para el *linear substitution calculus*, un cálculo ES a distancia. En particular, demostramos la unicidad de reducciones standard considerando al conjunto de los términos de este cálculo módulo una relación de equivalencia. Para obtener este resultado, desarrollamos una nueva demostración abstracta de estandarización en el modelo ARS, y demostramos que todas las nociones que intervienen en la representación de un sistema de reescitura en el modelo ARS, tales como paso de reducción, residuo, etc., son invariantes respecto de la relación de equivalencia mencionada.

Respecto de la **equivalencia entre reducciones**, presentamos una caracterización de la equivalencia por permutaciones sucesivas, para los sistemas de reescritura infinitarios lineales a izquierda de primer orden, mostrando que permite analizar distintos casos en los que resulta necesaria una cantidad infinita de permutaciones para mostrar la equivalencia entre dos reducciones. Para obtener esta caracterización, se extiende el modelo de *proof terms*, en la formulación para reescritura de términos de primer orden descripta en [BKdV03], al ámbito de la reescritura infinitaria. Demostramos que toda secuencia de reducción fuertemente convergente puede ser representada por un *proof term*, y además, que dicho *proof term* es único, módulo la asociatividad del operador binario de concatenación. Utilizamos el modelo de reescritura infinitaria mediante *proof terms* para presentar una demostración alternativa del resultado de compresión de reducciones fuertemente convergentes.

[4] En rigor el resultado obtenido es más general, aplica a cualquier *actividad de contracción*.

Como posibles líneas de **trabajo futuro**, mencionamos las siguientes.

Respecto del trabajo sobre **normalización**, estudiar si las ideas que subyacen a la definición de la estrategia de reducción para el *Pure Pattern Calculus* que presentamos en esta tesis, pudieran dar lugar a la definición de estrategias multipaso para otros sistemas de reescritura de alto orden, o mejor aún, para familias de dichos sistemas. Por otro lado, creemos que resulta interesante estudiar la posibilidad de extender la demostración abstracta de normalización que presentamos en esta tesis. Aunque no hemos logrado demostrar que la selección sistemática de conjuntos de pasos necesarios sea suficiente para demostrar que una estrategia de reducción es normalizante, prescindiendo así de la noción adicional de conjunto de pasos *non-gripping*, tampoco hemos encontrado ningún contraejemplo; en la opinión del autor, dilucidar esta cuestión implicaría avanzar un paso en la comprensión de las estrategias multipaso.

Respecto del trabajo sobre **estandarización**, evaluar la aplicabilidad de la idea de la aplicación de una operación *a distancia*, en el estudio del fenómeno de *pattern matching*, mediante la definición de *cálculos de matching explícito a distancia*. Aunque varios cálculos de matching explícito, basados en cálculos con patrones, han sido propuestos y estudiados en la literatura, el autor no conoce ningún estudio de estandarización que aplique a cálculos de matching explícito. Conjeturamos que un cálculo de matching explícito a distancia podría tener asociado un espacio de reducciones menos complejo, posibilitando de esta forma el estudio de resultados de estandarización para el mismo.

Respecto de la **equivalencia de reducciones para reescritura infinitaria**, creemos que algunas de las ideas subyacentes a la demostración del resultado de compresión que se presenta en esta tesis, pueden dar lugar al desarrollo de demostraciones genéricas de resultados de *estandarización* para reescritura infinitaria, utilizando el modelo de proof terms. Tal como se indica en [Ket12], se requiere una noción de paso *externo* que resulte adecuada a la reescritura infinitaria. Conjeturamos que utilizando la noción propuesta en ese trabajo, pueden obtenerse resultados de *unicidad* de reducciones standard. Otra dirección posible de trabajo futuro es la comparación de la caracterización de la equivalencia por permutaciones sucesivas que presentamos en esta tesis, con otras posibles caracterizaciones de la equivalencia entre reducciones infinitarias. Al respecto, destacamos que en [KKSdV95], y también en [BKdV03], Capítulo 12, se presenta una definición de equivalencia basada en proyecciones, que extiende la llamada "Lévy-equivalencia", cfr. [HL91]. Por otra parte, en [BKdV03], Capítulo 8, así como en [vOdV02], se demuestra la equivalencia entre varias caracterizaciones de la equivalencia entre reducciones, para reescritura finitaria.

Finalmente, mencionamos que en este capítulo, además de la reseña de los resultados obtenidos y la descripción de posibles direcciones de trabajo futuro, presentamos algunas notas relacionadas con el uso, en esta tesis, de los modelos genéricos de los *sistemas abstractos de reescritura* (ARS) y de *proof terms*. Estas notas ponen en relevancia los indicios que pueden proporcionar estos modelos genéricos para la comprensión de las características de distintos sistemas de reescritura, y brindan algunos elementos que comparan los dos modelos utilizados.

Destacamos cómo se refleja, en cada modelo, la noción de **ortogonalidad** en sistemas de reescritura de términos. El modelo ARS da una *caracterización semántica* de la ortogonalidad que permite tratar como ortogonales algunos sistemas de reescritura que no resultan tales de acuerdo a un criterio sintáctico, como es el caso del *linear substitution calculus*, o para los cuales resulta difícil analizar su ortogonalidad desde un punto de vista sintáctico, debido a la forma en que están definidos, como es el caso del *Pure Pattern Calculus*. Por su parte, el modelo de *proof terms* está *enfocado en la descripción detallada de reducciones individuales*. Notamos que para cualquier situación en la que la falta de ortogonalidad implica una elección entre opciones mutuamente incompatibles, en una reducción particular se elige, a lo sumo, una de estas opciones, pudiéndose establecer cuál es la opción elegida. Esta observación implica que la problemática de la falta de ortogonalidad pierde relevancia en el modelo de *proof terms*. Por lo tanto, en este modelo pueden obtenerse resultados que, en otros enfoques, quedarían restringidos a sistemas ortogonales, de forma tal que resulten válidos para familias de sistemas de reescritura que incluyan tanto sistemas ortogonales como no ortogonales.

Appendix B

Résumé en français

B.1 Introduction

On aborde dans cette thèse certaines propriétés formelles de systèmes de réécriture qui concernent leurs *espaces des dérivations*. Les systèmes de réécriture choisis présentent des caractéristiques particulières qui font l'étude des propriétés choisies des défis intéressants.

Dans la suite, on présente les systèmes étudiés dans cette thèse.

- Le chapitre 3 est dédié au *Pure Pattern Calculus*, PPC dans la suite. Il s'agit d'un *calcul avec motifs*. Un attribut clé de ce calcul est que l'ensemble des motifs est le même que celui des termes. Notamment, des pas de réduction peuvent être effectués à l'intérieur d'un motif; c'est à dire, les motifs sont *dynamiques*. Une opération de *matching* soigneusement définie permet de préserver la confluence dans le PPC.

  On étudie la question de l'existence de stratégies de calcul normalisantes pour le PPC. On remarque que le dispositif pour gérer les erreurs de matching implique son caractère *non-séquentiel*. Par conséquent, les résultats dérivés de la notion de radical nécessaire ne peuvent être appliqués pour le PPC.

- Le sujet du chapitre 4 est le *Linear Substitution Calculus*, $\lambda_{\mathtt{lsub}}^{\sim}$ dans la suite, un calcul appartenant à la famille des calculs avec substitutions explicites. La caractéristique la plus importante de ce calcul est qu'on peut appliquer une substitution explicite concernant une certaine variable, à une occurrence *distante* de cette variable, c'est à dire, une occurrence non juxtaposée à la substitution. Il existe une forte corrélation entre le $\lambda_{\mathtt{lsub}}^{\sim}$ et les réseaux de preuves utilisés dans la logique linéaire. Une relation d'équivalence $\sim$ dans l'ensemble des termes de $\lambda_{\mathtt{lsub}}^{\sim}$, entraînée par trois équations, permet d'établir un vrai isomorphisme: le comportement d'un réseau de preuve correspond exactement à celui de n'importe quel terme dans une certaine classe de $\sim$-équivalence.

  On établit des critères et des résultats de *standardisation* pour le Linear Substitution Calculus. Certains d'entre eux sont définis sur l'ensemble des termes modulo la relation $\sim$ mentionnée ci dessus.

- Dans le chapitre 5, on étudie les systèmes de *réécriture infinitaire* du premier ordre, linéaires à gauche. On adopte dans cette thèse le critère de convergence forte pour la définition des dérivations.

Plus précisément, on propose une caractérisation de l'*équivalence entre dérivations infinitaires*. On montre l'adéquation de notre définition dans plusieurs exemples. Notamment, on discute un phénomène que l'on trouve seulement dans la réécriture infinitaire: l'existence d'un type particulier d'effacement de sous-termes.

On utilise la notion de équivalence définie pour développer une preuve d'une version renforcée du résultat de *compression* des dérivations infinitaires.

Un trait commun aux trois sujets abordés dans ce travail est l'utilisation de *formalismes génériques* de systèmes de réécriture. Le matériel des chapitres 3 et 4 repose sur les *Systèmes Abstraits de Réécriture*, tels qu'ils sont décrits dans [Mel96]. De son côté, le chapitre 5 est fondé sur la notion de *proof term*. On étend à la réécriture infinitaire la formulation pour les systèmes de réécriture du premier ordre introduite dans [BKdV03], où les dérivations sont modélisés comme des *proof terms*. Une introduction aux deux modèles génériques utilisés fait le sujet du chapitre 2.

Dans cette thèse, on décrit chaque calcul abordé dans le cadre de l'un de ces deux modèles génériques. On se sert de la possibilité de développer des preuves abstraites dans ces modèles pour aboutir aux résultats désirés.

B.2 Modèles génériques de réécriture

On décrit dans ce chapitre les modèles génériques de réécriture qu'on utilisera dans la suite de cette thèse.

Le premier de ces modèles est celui qui repose sur la notion de *Système Abstrait de Réécriture*, dans la suite ARS, dû à la sigle en anglais.

Un ARS est défini comme une structure ayant la forme $\langle \mathcal{O}, \mathcal{R}, \mathsf{src}, \mathsf{tgt}, [\![\cdot]\!], < \rangle$, où $\mathcal{O}$ et $\mathcal{R}$ désignent deux ensembles de termes (ou objets) et radicaux respectivement; $\mathsf{src}, \mathsf{tgt} : \mathcal{R} \to \mathcal{O}$ modélisent les termes de départ et d'arrivée de chaque radical, $[\![\cdot]\!] \subseteq \mathcal{R} \times \mathcal{R} \times \mathcal{R}$ est la relation de résidus, et $< \subseteq \mathcal{R} \times \mathcal{R}$ est la relation d'emboîtement.

La preuve abstraite de normalisation développée dans le chapitre 3 se sert d'une version étendue de la définition de ARS, qui inclut une relation additionnelle $\ll \subseteq \mathcal{R} \times \mathcal{R}$ dit d'agrippement.

Diverses caractéristiques d'un calcul peuvent être modélisés dans le cadre des ARS sous la forme d'*axiomes*.

À titre d'exemple, la finitude des développements et l'orthogonalité sont décrits par des axiomes. À propos du dernier, nous soulignons que l'orthogonalité est définie sur la base de la relation abstraite de résidus, produisant une description ayant un caractère plus sémantique que celle fondée sur la notion de paire critique.

Les traits les plus marquants de la relation d'emboîtement se modélisent également comme des axiomes. Nous évoquons l'axiome dit de *linéarité*:

$$a \not< b \;\Leftrightarrow\; \exists! b'. b[\![a]\!]b'$$

qui définit une condition, liée à l'emboîtement, qui doit satisfaire un radical a pour avoir la capacité de multiplier, ou bien d'effacer, un autre radical b.

Dans ce modèle de réécriture, une réduction est définie simplement comme une séquence de radicaux $r_0, r_1, \ldots$ qui vérifie $\mathsf{tgt}(r_i) = \mathsf{src}(r_{i+1})$ pour tout i. Étant donné un ensemble de radicaux co-initiaux $\mathcal{A}$, un *développement* de $\mathcal{A}$ est n'importe quelle réduction $r_0, r_1, \ldots$ telle que pour tout i, $r_i \in \mathcal{A}[\![r_0]\!][\![r_1]\!] \ldots [\![r_{i-1}]\!]$, où $\mathcal{A}[\![r]\!]$ est défini comme $\bigcup_{a \in \mathcal{A}} \{b \,/\, a[\![r]\!]b\}$. Un développement $r_0, r_1, \ldots, r_n$ est *complet* si $\mathcal{A}[\![r_0]\!][\![r_1]\!] \ldots [\![r_n]\!] = \varnothing$. On peut montrer de manière abstraite que pour n'importe quel ARS qui vérifie les axiomes d'orthogonalité et de finitude des développements, en même temps que d'autres axiomes basiques, tous les développements complets d'un ensemble de radicaux co-initiaux terminent sur le même terme et induisent la même relation de résidus.

Ce résultat permet de décrire, de manière simple, la notion d'équivalence de réductions par permutation de pas dans le cadre abstrait fourni par les ARS. Si d_1 et d_2 sont des réductions, alors $d_1; a; f; d_2$ et $d_1; b; e; d_2$ sont équivalentes à une permutation près lorsque e et f sont des développements complets de $a[\![b]\!]$ et $b[\![a]\!]$ respectivement. L'équivalence entre réductions se définit comme la clôture reflexive-transitive de cet relation. On peut définir un ordre de standardisation entre réductions équivalentes, en ayant recours à la relation d'emboîtement: on dit que $d_1; a; f; d_2$ est plus standard que $d_1; b; e; d_2$ si $a < b$ lorsque $a < b$.

La notion de *réduction standard* est precisée dans [Mel96] sur la base de cet ordre de standardisation. En outre, plusieurs résultats de standardisation, concernant l'existence ou l'unicité des réductions standards dans chaque classe de réductions équivalentes, sont énoncés et prouvés. Les conditions requises sur un ARS pour assurer ces résultats, sont décrits sous la forme d'axiomes.

L'idée de se servir des termes pour répresenter des réductions, donne lieu au deuxième modèle générique de réécriture qu'on utilise dans cette thèse. On appelle *proof terms* les termes qui répresentent des réductions. Dans la suite, nous décrivons le modèle fondé sur la notion de *proof term*, tel qu'introduit dans [BKdV03] pour les systèmes de réécriture des termes du premier ordre (TRS) linéaires à gauche. Ladite version du modèle est la base pour le materiel du chapitre 5 de cette thèse.

Un *proof term* pour un TRS T est un terme sur une signature qui étend celle de T. Pour chaque règle en T, on ajoute un symbole dont l'arité est le nombre de variables qui apparaissent dans la règle. Par exemple, les règles $f(x) \to g(x)$ et $g(x) \to k(x)$ donnent lieu à deux symboles unaires, disons μ et ν respectivement, dans la signature des *proof terms* pour tout TRS incluant ces règles. La signature des *proof terms* se complète par un symbole binaire, noté par le point, qui désigne la concaténation. Ainsi, la réduction $f(a) \to g(a) \to k(a)$ est dénoté par le *proof term* $\mu(a) \cdot \nu(a)$.

Nous soulignons que la notion de *proof term* donne des désignations particuliers pour les *réductions simultanées* de ensembles de radicaux. C'est à dire, la réduction simultanée d'un certain ensemble de radicaux est désignée par un *proof term*, qui est différent de ceux qui désignent n'importe quel autre option pour la réduction séquentielle des mêmes radicaux. Par exemple, le *proof term* $\mu(\mu(a))$, qui désigne la réduction simultanée des deux radicaux dans le terme $f(f(a))$, est différente soit de $\mu(f(a)) \cdot g(\mu(a))$ soit de $f(\mu(a)) \cdot \mu(g(a))$, qui dénotent les deux possibilités pour réduire les mêmes radicaux de façon séquentiel.

Dans ce modèle de réécriture, un schèma de logique equationelle, opérant sur des proof terms, permet de répresenter la notion de permutation de pas de réduction, et par conséquent, l'équivalence entre réductions. La base pour cette répresentation se compose des six schèmas d'équation suivantes

$$
\begin{array}{lrcl}
(\mathsf{IdLeft}) & src(\psi) \cdot \psi & \sim & \psi \\
(\mathsf{IdRight}) & \psi \cdot tgt(\psi) & \sim & \psi \\
(\mathsf{Assoc}) & \psi \cdot (\phi \cdot \chi) & \sim & (\psi \cdot \phi) \cdot \chi \\
(\mathsf{Struct}) & f(\psi_1, \ldots, \psi_m) \cdot f(\phi_1, \ldots, \phi_m) & \sim & f(\psi_1 \cdot \phi_1, \ldots, \psi_m \cdot \phi_m) \\
(\mathsf{OutIn}) & \mu(\psi_1, \ldots, \psi_m) & \sim & \mu(s_1, \ldots, s_m) \cdot r[\psi_1, \ldots, \psi_m] \\
(\mathsf{InOut}) & \mu(\psi_1, \ldots, \psi_m) & \sim & l[\psi_1, \ldots, \psi_m] \cdot \mu(t_1, \ldots, t_m)
\end{array}
$$

où s_i et t_i désignent les termes de départ et d'arrivée, respectivement, du *proof term* ψ_i; tandis que la règle désignée par μ a la forme $l[x_1, \ldots, x_m] \to r[x_1, \ldots, x_m]$. Par exemple, l'équivalence entre les réductions $h(f(a), f(a)) \to h(f(a), g(a)) \to h(g(a), g(a))$ et $h(f(a), f(a)) \to h(g(a), f(a)) \to h(g(a), g(a))$, où chacune de ces réductions est le résultat d'une permutation de pas parallèles sur l'autre, peut être établie moyennant leur répresentation comme *proof terms* comme suit:

$$
\begin{array}{rcll}
h(f(a), \mu(a)) \cdot h(\mu(a), g(a)) & \approx & h(f(a) \cdot \mu(a), \mu(a) \cdot g(a)) & (\mathsf{Struct}) \\
& \approx & h(\mu(a), \mu(a)) & (\mathsf{IdRight}), (\mathsf{IdLeft}) \\
& \approx & h(\mu(a) \cdot g(a), f(a) \cdot \mu(a)) & (\mathsf{IdLeft}), (\mathsf{IdRight}) \\
& \approx & h(\mu(a), f(a)) \cdot h(g(a), \mu(a)) & (\mathsf{Struct})
\end{array}
$$

L'équivalence entre réductions concernant la permutation de pas emboîtés peut être établie à l'aide des équations (OutIn) et (InOut). Par exemple, nous obtenons l'équivalence des réductions $f(f(a)) \to f(g(a)) \to g(g(a))$ et $f(f(a)) \to g(f(a)) \to g(g(a))$ comme suit:

$$
f(\mu(a)) \cdot \mu(g(a)) \approx \mu(\mu(a)) \approx \mu(f(a)) \cdot g(\mu(a))
$$

où l'on fait appel d'abord à (InOut), et après à (OutIn).

B.3 Normalisation

Le but général du chapitre 3 est d'atteindre des stratégies de réduction normalisantes et effectives pour le *Pure Pattern Calculus* en particulier, et pour des calculs non-séquentiels en général. Dans ce cadre, nous nous penchons sur des *stratégies "multi-radicaux"*, c'est à dire, celles qui permettent la sélection de plusieurs radicaux dans le même terme. Le terme suivant dans la dérivation donnée par une stratégie "multiradicaux" est le résultat de la réduction simultanée des radicaux choisis.

On peut distinguer deux parties dans ce chapitre.

Le matériel dans la première de ces parties est de nature *abstraite*. On travaille avec les *Systèmes Abstraites de Réécriture*, bref *ARS*, comme indiqué dans l'introduction. Dans ce cadre, nous présentons une preuve abstraite de normalisation inédite, orientée vers les stratégies "multiradicaux". Pour développer cette preuve, on se sert de la capacité des ARS pour modéliser les dérivations dans lesquelles chaque pas, ou chaque étape, correspond à la réduction simultanée d'un ensemble de radicaux co-initiaux.

Plus précisément, nous prouvons que, pour tout calcul satisfaisant un certain ensemble de conditions, la réduction d'*ensembles* de radicaux *nécessaires* et *non-agrippants* est normalisante.

Les conditions imposées sur le calcul incluent tous les axiomes requis dans les preuves de standardisation présentées dans le chapitre 4 de [Mel96], sauf pour l'un d'entre eux. L'axiome omis est celui dénommé *Stabilité*, qui décrit une condition liée au caractère séquentiel d'un calcul. Par conséquent, l'exclusion de cet axiome permet d'utiliser le cadre des ARS pour aboutir à des résultats que l'on peut appliquer sur des calculs non-séquentiels. D'autre part, la preuve que nous présentons fait recours à un nouvel axiome, que nous introduisons dans cette thèse. Cet axiome, qui nous appelons *Pivot*, complète l'analyse de la préservation de l'emboîtement dans les résidus, décrit par les axiomes de *non-contextualité* et d'*enclave*.

La notion d'*ensemble nécessaire de radicaux co-initiaux* étend celui de radical nécessaire. Un ensemble $\mathcal{A}$ de radicaux dans un terme t est dit *nécessaire* si toute dérivation de t vers une forme normale inclus la contraction de, au minimum, un radical dans $\mathcal{A}$, ou bien, d'un de ses résidus. Une preuve de normalisation présentée dans [SR93] établit que pour les systèmes de réécriture de termes du premier ordre, la réduction des ensembles nécessaires de radicaux est normalisante. Les idées principales de ce travail sont revisitées dans la preuve que nous développons. La généralisation de ces idées au cadre abstrait des ARS, dans lequel on peut modéliser des calculs d'ordre supérieur, requiert l'introduction des notions additionnelles.

Entre ces notions, on remarque que la preuve abstraite de normalisation décrite dans cette thèse fait appel à une version étendue du formalisme des ARS, incluant la relation d'*agrippement* entre radicaux. Cette relation est introduite dans [Mel96] pour donner une preuve abstraite de la finitude des développements. La condition d'être *non-agrippante*, qui exige des ensembles de radicaux réduits, est fondée sur cette relation. On doit également ajouter trois axiomes additionnelles, qui décrivent l'agrippement de forme abstraite, aux conditions imposées aux calcul.

Dans la deuxième partie de ce chapitre, nous appliquons au *Pure Pattern Calculus*, PPC dans la suite, le résultat abstrait de normalisation déjà décrit. Comme indiqué dans l'introduction, le PPC est un calcul avec motifs non-séquentiel, dans lequel les motifs sont dynamiques. Nous définissons une stratégie de réduction multiradicaux pour ce calcul, et nous prouvons que cette stratégie est normalisante, en faisant recours à la preuve abstraite développée dans le cadre des ARS.

On utilisera dans ce résumé la version simplifiée de la syntaxe du PPC qui suit:

$$t ::= x \mid \mathtt{c} \mid \lambda_\theta t.t \mid tt$$

où $\mathtt{c}$ désigne un élément d'un ensemble de constants, et θ est l'ensemble de variables liées par la construction d'abstaction. Par exemple, les trois occurrences de la variable z dans le terme $\lambda_{\{z,w\}} zw.(\lambda_{\{x,y\}} zxy.zyx)w$ sont liées par l'abstraction extérieure, même celles incluses dans l'abstraction interne. On désignera la fonction identité, i.e. $(\lambda_{\{x\}} x.x)$, avec la lettre I.

La sémantique opérationnelle du PPC est définie par la règle suivante, qui généralise la règle β du λ-calcul classique:

$$(\lambda_\theta p.s)u \to \{p/_\theta\ u\}s \qquad \text{si } \{p/_\theta\ u\} \text{ est } \textit{décidé}$$

La notation $\{p/_\theta\ u\}$ désigne l'opération de *filtrage* du motif p avec l'argument u, en relation à l'ensemble de variables θ. Étant donnés deux termes p et u, et un ensemble θ, il y a trois issues possibles pour l'opération de filtrage:

1. *filtrage positif*: l'argument se conforme au motif. Dans ce cas, le résultat du filtrage est une substitution dont le domaine est θ. Par exemple, $\{\mathtt{a}x/_{\{x\}}\ \mathtt{a}(I\mathtt{c})\} = \{x := I\mathtt{c}\}$.

2. *filtrage négatif*: la forme de l'argument est différente de celle du motif. Dans ce cas, on obtient la valeur bien connue `fail` comme résultat du filtrage. Par exemple, $\{\mathtt{a}x/_{\{x\}}\ \mathtt{b}(I\mathtt{c})\} = \mathtt{fail}$.

3. *filtrage non décidé*: l'application de pas de réduction internes au motif, á l'argument, ou aux deux, est nécessaire pour aboutir á des termes pour lesquels on peut décider si le filtrage est positif ou négatif; la décision n'est pas possible pour les termes donnés. Dans ce cas, le résultat est la valeur bien connue `wait`. Par exemple, $\{\mathtt{a}x/_{\{x\}}\ I(\mathtt{ac})\} = \{\mathtt{a}x/_{\{x\}}\ I(\mathtt{bc})\} = \mathtt{wait}$. Dans l'un ou l'autre cas, un pas de réduction dans l'argument permet d'obtenir un résultat décidé: $\{\mathtt{a}x/_{\{x\}}\ \mathtt{ac}\} = \{x := \mathtt{c}\}$, $\{\mathtt{a}x/_{\{x\}}\ \mathtt{bc}\} = \mathtt{fail}$.

Un filtrage est dit *décidé* s'il est positif ou négatif.

Nous remarquons qu'un terme ayant la forme $(\lambda_\theta p.s)u$ dont le filtrage $\{p/_\theta\ u\}$ est non décidé, ne correspond pas à un radical du `PPC`. Par exemple, le seul radical dans le terme $(\lambda_{\{x\}}\mathtt{a}x.x)(I(\mathtt{ab}))$ est celui correspondant au sous-terme $I(\mathtt{ab})$. On désigne comme *pré-radical* chaque sous-terme ayant la forme $(\lambda_\theta p.s)u$ dans un terme, qu'il s'agisse d'un radical ou pas.

On complète la définition du `PPC` en indiquant que l'application de la valeur `fail` à n'importe quel terme rapporte le terme I, c'est-à-dire, la fonction identité I. Par conséquent, on a $(\lambda_{\{x\}}\mathtt{a}x.\mathtt{d}x)(\mathtt{ac}) \to \mathtt{dc}$ et $(\lambda_{\{x\}}\mathtt{a}x.\mathtt{d}x)(\mathtt{bc}) \to I$.

À propos du filtrage, on souligne que dans le cas où soit le motif soit l'argument sont des termes composés, l'échec de filtrage (c'est-à-dire, la constatation d'une différence entre le motif et l'argument) dans n'importe quel composante, entraîne que le filtrage composé est négatif. Par exemple, $\{\mathtt{abc}x/_{\{x\}}\ \mathtt{adce}\} = \{\mathtt{abc}x/_{\{x\}}\ \mathtt{abde}\} = \mathtt{fail}$. Le caractère non-séquentiel du `PPC` provient de cette particularité. À titre d'exemple, on considère le terme $t = (\lambda_{\{x\}}\mathtt{abc}x.x)(\mathtt{a}(I\mathtt{d})(I\mathtt{d})\mathtt{e})$. Aucun des deux radicaux de ce terme est nécessaire, comme indiqué par les réductions $t \to (\lambda_{\{x\}}\mathtt{abc}x.x)(\mathtt{ad}(I\mathtt{d})\mathtt{e}) \to I$ et $t \to (\lambda_{\{x\}}\mathtt{abc}x.x)(\mathtt{a}(I\mathtt{d})\mathtt{de}) \to I$. Le fait qu'il suffit de réduire n'importe lequel des deux radicaux de t pour atteindre la forme normale I, est dû au fait que la différence entre `b` et `d`, ou celle entre `c` et `d`, suffit pour obtenir un filtrage négatif, et par conséquent décidé, avec le motif $\mathtt{abc}x$.

La stratégie de réduction que nous définissons dans ce chapitre se concentre sur le pré-radical plus extérieure – plus à gauche (dans la suite LO, dû à l'acronyme anglais pour "leftmost-outermost"). Si le filtrage qui correspond à ce pré-radical est décidé, c'est-à-dire, si ce pré-radical est en fait un radical, alors le choix se porte sur ce radical uniquement. Autrement dit, le choix se porte sur des radicaux internes au motif et/ou à l'argument du pré-radical LO. Dans ce cas, on cherche particulièrement des radicaux que peuvent contribuer á l'obtention d'un terme dans lequel le radical LO soit décidé.

Le point concernant le filtrage qui rend le `PPC` non-séquentiel, détermine aussi l'existence des termes pour lequels la stratégie doit choisir la réduction simultané de plusieurs radicaux coinitiaux, et par conséquent, le caractère *multiradicaux* de cette stratégie. Par exemple, pour n'importe lequel des termes suivantes: $(\lambda_{\{x\}}\mathtt{abc}x.x)(\mathtt{a}(I\mathtt{b})(I\mathtt{d})\mathtt{e})$, $(\lambda_{\{x\}}\mathtt{abc}x.x)(\mathtt{a}\Omega(I\mathtt{d})\mathtt{e})$, $(\lambda_{\{x\}}\mathtt{abc}x.x)(\mathtt{a}(I\mathtt{d})(I\mathtt{c})\mathtt{e})$, et $(\lambda_{\{x\}}\mathtt{abc}x.x)(\mathtt{a}(I\mathtt{d})\Omega\mathtt{e})$, où Ω désigne un terme non-normalisant, la stratégie doit choisir la réduction simultané des deux radicaux internes à l'argument. Ce choix est nécessaire pour atteindre, dans tous les cas, un terme dont le filtrage du pré-radical exterieur est décidé.

La concentration sur le pré-radical LO et, en plus, la façon dont les radicaux à réduire son choisis si ce pré-radical est non décidé, portent sur une stratégie judicieuse, dans le sens de limiter le nombre des cas dans lesquels plus d'un radical est choisi, et plus en général, le nombre de radicaux choisis pour chaque terme. Nous remarquons que, notamment, cette stratégie s'accorde avec la stratégie "plus extérieur – plus à gauche", choisissant exactement un radical pour chaque terme, si on considère la restriction du PPC à l'ensemble des termes avec correspondance immédiate avec le λ-calcul classique.

Pour vérifier que la stratégie décrite est normalisante, nous définissons un ARS qui modèle le PPC. Notamment, la définition de cet ARS inclut la relation d'agrippement. Nous prouvons que cet ARS vérifie tous les axiomes requis dans la preuve abstraite de normalisation développée dans la première partie de ce chapitre, et également, que cette stratégie aboutit toujours à la réduction d'ensembles de radicaux nécessaires et non-agrippantes.

B.4 Standardisation pour le *linear substitution calculus*

L'objectif de ce chapitre est d'obtenir des résultats de standardisation pour le *linear substitution calculus*, $\lambda_{\mathtt{lsub}}^{\sim}$ dans la suite, en utilisant le modèle de réécriture défini par la notion de ARS.

Comme nous avons signalé dans l'introduction, $\lambda_{\mathtt{lsub}}^{\sim}$ est un calcul avec substitutions explicites (ES), ayant la capacité d'agir *à distance*. Une brève description du calcul permet d'observer cette caractéristique.

La syntaxe de $\lambda_{\mathtt{lsub}}^{\sim}$ est définie ainsi:

$$t ::= x \mid \lambda x.t \mid tt \mid t[x/t]$$

On utilisera $\mathtt{L}$, $\mathtt{L}'$, etc., pour désigner des listes de substitutions ayant la forme $[x_1/t_1]\dots[x_n/t_n]$. Nous soulignons qu'une liste de substitutions, toute seule, n'est pas un terme.

La sémantique de $\lambda_{\mathtt{lsub}}^{\sim}$ est donnée par les trois règles de réduction suivantes:

$(\lambda x.t)\mathtt{L}u$	$\to_{\mathtt{db}}$	$t[x/u]\mathtt{L}$		béta-à-distance
$C[\![x]\!][x/u]$	$\to_{\mathtt{ls}}$	$C[\![u]\!][x/u]$		substitution linéaire
$t[x/u]$	$\to_{\mathtt{gc}}$	t	if $x \notin \mathtt{fv}(t)$	ramasse-miettes

où $[\![x]\!]$ dénote une occurrence libre de x dans le contexte C. La forme de la règle de substitution linéaire permet, comme nous avons remarqué dans l'introduction, d'appliquer une substitution à une occurrence de variable qui n'est pas adjacente à la substitution en question. Par exemple, dans

$$((xy)(xz))[x/w] \to (wy)(xz)[x/w] \to (wy)(wz)[x/w]$$

deux pas de réduction, correspondant aux deux occurrences libres de x dans $(xy)(xz)$, suffisent pour appliquer la substitution. D'ailleurs, la règle béta-à-distance permet d'appliquer une abstraction à un argument dont elle est séparée par une liste de substitutions. De cette manière, on évite la nécessité de multiplier et de déplacer les substitutions explicites dans le terme, en obtenant ainsi un espace de réductions plus simple que celui d'autres calculs avec ES.

Le caractère "à distance" du calcul $\lambda^{\sim}_{\texttt{lsub}}$ fait que l'emplacement précis d'une substitution explicite soit parfois négligeable. Cette remarque témoigne l'analogie entre les calculs ES à distance et les réseaux de preuve introduites dans le cadre de la logique linéaire. La relation d'équivalence dans l'ensemble de termes définie par les équations suivantes

$$\begin{array}{llll} t[x/u][y/s] & \approx_{\texttt{CS}} & t[y/s][x/u] & x \notin \texttt{fv}(s) \ \& \ y \notin \texttt{fv}(u) \\ (\lambda y.t)[x/u] & \approx_{\sigma_1} & \lambda y.t[x/u] & y \notin \texttt{fv}(u) \\ (ts)[x/u] & \approx_{\sigma_2} & t[x/u]s & x \notin \texttt{fv}(s) \end{array}$$

permet d'établir un isomorphisme entre les classes d'équivalence de termes et les réseaux de preuve. On appelle $\sim$ cette relation.

Une première définition d'un ARS qui modélise $\lambda^{\sim}_{\texttt{lsub}}$ permet d'obtenir des premiers résultats de standardisation pour ce calcul.

On signale que ce modèle permet d'établir une distinction entre des radicaux différentes, déterminés par la règle de substitution linéaire, qui correspondent au même sous-terme d'un terme donné, comme c'est le cas des deux radicaux du terme $(xx)[x/y]$.

La relation d'emboîtement pour cet ARS est un ordre "gauche vers droite", dénoté par $\prec_{\texttt{L}}$ et défini comme suit: étant donnés deux radicaux co-initiaux r_1 et r_2, on définit $r_1 \prec_{\texttt{L}} r_2$ si l'*ancre* de r_1 est, textuellement, à gauche de celle de r_2, où l'ancre d'un radical est l'expression soulignée comme suit pour chaque règle: $(\underline{\lambda x}.s)\texttt{L}u$, $C[\![\underline{x}]\!][x/u]$, $t[\underline{x}/u]$. Cet définition produit un ordre total, ce qui simplifie la vérification de certains axiomes requis dans les résultats de standardisation présentés dans [Mel96]. L'application d'étiquettes sur l'ancre de chaque radical, et l'observation du comportement de cettes étiquettes, permettent de définir la relation de résidus du ARS qui modèlisent $\lambda^{\sim}_{\texttt{lsub}}$.

L'ARS ainsi défini permet d'arriver à une caractérisation forte de la notion de reduction standard pour $\lambda^{\sim}_{\texttt{lsub}}$: chaque classe de reductions équivalentes inclut, exactement, une reduction standard.

D'autre part on remarque que l'ordre $\prec_{\texttt{L}}$ n'est pas invariant par rapport à la relation d'équivalence $\sim$ mentionée plus haut. Par exemple, si l'on considère les termes

$$t_1 = (x\underline{y})[x/\underline{w}[w/z]][y/y'] \sim (x[x/\underline{w}[w/z]])(\underline{y}[y/y']) = t_2$$

on voit que le radical correspondant à l'occurrence soulignée de y précède, dans t_1, celui de l'occurrence soulignée de w, tandis que cet ordre devient inverse dans t_2.

Pour décrire $\lambda^{\sim}_{\texttt{lsub}}$ comme un calcul de réécriture opérant sur des classes d'équivalence de termes par rapport à la relation $\sim$, on définit d'abord une bijection entre les ensembles de radicaux de termes équivalents, fondée sur l'étiquetage décrit auparavant. Nous montrons que cette bijection établit une bisimulation entre les radicaux, par rapport à la relation de résidus.

Par ailleurs, pour resoudre le problème mentionné concernant l'emboîtement gauche-vers-droite, on définit un deuxième ARS, dont la relation d'emboîtement, que nous baptisons "ordre de boite", et notée $\prec_{\texttt{B}}$, est un sous-ordre strict de $\prec_{\texttt{L}}$.

Tandis que ce deuxième ARS capture de manière adéquate la notion de réduction modulo, d'après ce qu'on vient de dire, il ne satisfait pas tous les axiomes requis dans [Mel96] pour obtenir un résultat fort de standardisation. Pour arriver à un tel résultat, nous développons une nouvelle preuve abstraite de standardisation dans le cadre des ARS, laquelle s'applique, en effet, au deuxième modèle obtenu de $\lambda^{\sim}_{\texttt{lsub}}$.

Il est important de signaler que le fait de resortir au cadre abstrait pourvu par la notion de ARS permet d'obtenir des résultats intéressants pour un calcul, tel le $\lambda^{\sim}_{\mathtt{lsub}}$, dont l'espace de réductions doit être considéré modulo une relation d'équivalence sur l'ensemble de termes.

B.5 Équivalence de réductions pour la réécriture infinitaire

Le but de ce chapitre est de transférer la description de l'espace des réductions des calculs fondée sur le concept de *proof term*, telle qu'elle est définie dans [BKdV03] pour les systèmes de réécriture des termes (TRS) linéaires à gauche, au cadre de la réécriture infinitaire. La définition des réductions infinitaires tiendra compte du critère de convergence forte.

On trouve deux défis par rapport à la tâche de modéliser les réductions infinitaires: d'une part on doit décrire des réductions de termes infinis et, d'autre part, il faut décrire (et raisonner sur) des réductions ayant une *longueur* infinie, notamment des réductions dont la longueur dépasse le premier ordinal infini ω.

On est obligé donc de travailler avec des termes infinis, par exemple, si nous donnons le nom μ à la règle $f(x) \to g(x)$, on peut considérer le *proof term* $\mu(f^\omega)$, qui désigne le pas de réduction $f^\omega = f(f^\omega) \to g(f^\omega)$, où $f^\omega = f(f(\ldots))$ et le symbole f apparaît ω fois.

En outre, le deuxième des deux défis signalés pose le problème de trouver une manière adéquate de modéliser la concaténation d'un nombre infini de pas de réduction ou bien, des réductions en général.

Étant donnés ces remarques, nous définissons dans ce chapitre l'ensemble des proof termes infinitaires, de même que leurs termes de départ (source) et d'arrivée (target). Nous caractérisons formellement la profondeur minimale de (la réduction décrite par) un *proof term.*

La définition de l'ensemble des *proof terms* inclut deux règles différentes pour la concaténation: l'une pour la concaténation binaire, servant à définir le *proof term* $\psi \cdot \phi$ où ψ et ϕ sont des *proof terms*, l'autre pour la concaténation infinie, notée $\cdot_{i<\omega}\ \psi_i$, où chaque ψ_i est un *proof term.* Par exemple, en utilisant la règle μ définie plus haut, on peut considérer le *proof term* $\cdot_{i<\omega}\ f^i(\mu(g^\omega))$ qui dénote la réduction infinie $f^\omega \to g(f^\omega) \to g(g(f^\omega)) \twoheadrightarrow g^\omega$. La propriété de convergence d'une réduction infinitaire admet une caractérisation simple dans le cadre des *proof terms*; celle-ci repose sur la notion de profondeur minimale des *proof terms.*

La définition des *proof terms* est simultanée avec celle d'un ordinal dénombrable associé de manière unique à chaque *proof term.* Cette association permet de faire appel à l'induction transfinie pour donner une définition précise de l'ensemble des *proof terms* et pour raisonner sur cet ensemble. Dans la dite association, les ordinaux limite se correspondent exactement avec les concaténations infinies.

Nous étendons de même la caractérisation de l'équivalence entre réductions décrite dans [BKdV03], Sec. 8.3, aux réductions infinitaires. Nous rappelons que cette caractérisation se base sur l'application de la logique équationelle aux *proof terms*, à partir d'un ensemble de schémas d'équation basiques qui modélisent la permutation des pas de réduction adjacents.

Pour adapter cette caractérisation à la réécriture infinitaire, nous ajoutons une règle qui permet de faire appel à la notion de limite dans un raissonement équationel. La structure d'une telle règle est la suivante: étant donnés deux *proof terms* ψ et ϕ, si l'on peut construire deux séquences $\langle\psi_i\rangle_{i<\omega}$ et $\langle\phi_i\rangle_{i<\omega}$ telles que ψ et ϕ peuvent être montrés équivalents à ψ_i et ϕ_i, respectivement, pour tout i, et en plus, la limite des distances entre ψ_i et ϕ_i est nulle quand i tend vers l'infini, alors on conclut que ψ et ϕ sont, eux-mêmes, équivalents.

En ajoutant la règle que nous venons de décrire, en même temps qu'une règle de congruence et un schéma d'équation basique en relation avec la concaténation infinie, on obtient un système de logique équationelle adéquat pour raisonner sur l'équivalence des (*proof terms* qui désignent des) réductions infinitaires. En utilisant ce système on a pu vérifier l'équivalence des réductions dans plusieurs exemples impliquant des réductions infinies. Notamment, cette caractérisation de l'équivalence offre un modèle adéquat, à notre avis, du phénomène de l'*effacement infinitaire*, dans lequel une partie d'un réduction est effacée suite à une chaîne infinie de permutations de pas de réduction, tandis que l'effacement ne se produit pas après n'importe quel préfixe fini de cet chaîne.

La caractérisation des réductions infinitaires fournie par les *proof terms* est complète: nous montrons que toute réduction infinitaire convergente peut être représentée d'une manière précise par un *proof term*. On montre aussi que cette représentation est unique modulo associativité de la concaténation, c'est à dire, en identifiant, par exemple, $\psi \cdot (\phi \cdot \chi)$ et $(\psi \cdot \phi) \cdot \chi$.

Pour montrer la complétude et l'unicité de la représentation des réductions, on définit le sous-ensemble des *proof terms* séquentiels (stepwise proof terms) qui décrivent des réductions dont les pas sont réalisés en forme strictement séquentielle. Par exemple, le *proof term* $\mu(f(a)) \cdot g(\mu(a))$ est séquentiel, tandis que $\mu(\mu(a))$ ne l'est pas. Nous précisons aussi la notion d'"égalité modulo associativité de la concaténation" dans le contexte de la réécriture infinitaire, en utilisant le même schéma de logique équationelle défini pour modéliser l'équivalence entre réductions, où l'on prend l'associativité de la concaténation comme seul schéma d'équation basique.

Finalement, nous donnons une preuve du résultat connu sous le nom de *compression*, c'est à dire, que si l'on se restreint à des réductions de longueur au plus ω, on ne perd pas le pouvoir expressif de la réécriture infinitaire car, pour n'importe quelle réduction convergente ψ, il existe une autre équivalente ψ' de longueur finie ou ω. L'énoncé que nous montrons est, en fait, plus fort que ceux que nous avons trouvés dans la littérature, étant donné qu'on établit l'équivalence, et non seulement la coincidence des termes de départ et d'arrivée, entre la réduction originale et sa "comprimée" correspondante. En plus, on ne demande pas que le système de réécriture sous-jacent soit orthogonal, comme c'est le cas de la preuve donnée en [KKSdV95].

La preuve que nous présentons est basée sur le fait suivant: étant donnée une réduction dénotée par le *proof term* ψ, et un entier naturel n, on peut construire une forme factorisée de ψ, c'est à dire, un *proof term* $\psi' = \psi_1 \cdot \psi_2$ de sorte que ψ est équivalent à ψ', ψ_1 désigne une réduction finie, et en plus, la profondeur minimale de ψ_2 est supérieure à n.

B.6 Conclusions

Comme indiqué dans l'introduction de ce résumé, on aborde dans cette thèse l'étude de propriétés formelles de plusieurs calculs de réécriture. Plus précisément, on obtient des résultats concernant la normalisation pour le *Pure Pattern Calculus* `PPC`; la standardisation pour le *Linear Substitution Calculus* $\lambda^{\sim}_{\mathtt{lsub}}$; et l'équivalence entre réductions pour des systèmes de réécriture infinitaires du premier ordre.

Tous les calculs considérés présentent des caractéristiques qui rendent difficile l'étude des propriétés traitées dans cette thèse. Dans la suite, nous donnons un bref aperçu de ces caractéristiques.

- Par rapport au `PPC`, nous rappelons que l'échec de filtrage dans n'importe quel composant d'un filtrage composé entraîne le fait que ce filtrage soit négatif. Cela détermine soit le caractère non-séquentiel du `PPC`, soit la nécessité de définir une stratégie multiradicaux afin d'obtenir des résultats de normalisation.

- Dans le cas de $\lambda^{\sim}_{\mathtt{lsub}}$, le manque d'adéquation de l'emboîtement syntaxique pour pouvoir être considéré comme un ordre de standardisation entre radicaux coinitiaux, est dû à deux facteurs.

  L'un de ces facteurs est que l'emboîtement syntaxique ne vérifie pas la condition exprimée par l'axiome de linéarité: il se peut que deux radicaux a et b soient tels que a multiplie ou efface b, et a ne contient pas syntaxiquement b. Par exemple, si l'on considère $t = \underline{x}[x/\underline{y}][y/z]$, et que l'on appelle a_x et a_y les radicaux correspondant aux occurrences sur-lignés de x et de y respectivement, on observe que la réduction de a_x résulte en $t' = \underline{y}[x/\underline{y}][y/z]$, et donc, provoque une duplication du radical a_y, tandis que c'est le radical a_y celui qui contient a_x dans le terme t.

  L'autre facteur est que l'emboîtement syntaxique n'est pas invariant par rapport à la relation d'équivalence qui permet d'etablir un isomorphisme avec les réseaux de preuve.

- Par rapport à la réécriture infinitaire nous soulignons que, afin de définir l'équivalence entre réductions sur la base de la notion de permutation, on doit tenir compte du fait que, dans certains cas, un nombre infini de permutations est nécessaire pour obtenir l'équivalence entre deux réductions. Notamment, c'est possible qu'il soit nécessaire de permuter un ensemble infini de pas, et chacun d'entre eux, par rapport à un autre ensemble infini de pas.

  Par exemple, si l'on considère les règles $f(x) \to g(x)$ et $g(x) \to k(x)$, pour obtenir que la réduction $f^\omega \to g(f^\omega) \to g(g(f^\omega) \twoheadrightarrow g^\omega \to k(g^\omega) \to k(k(g^\omega)) \twoheadrightarrow k^\omega$ est équivalente à $f^\omega \to g(f^\omega) \to k(f^\omega) \to k(g(f^\omega)) \to k(k(f^\omega)) \twoheadrightarrow k^\omega$, on doit permuter tous les pas correspondentes à la règle $g(x) \to k(x)$, c'est à dire, un nombre infini de pas. D'ailleurs, chacun de ces pas doit être permuté avec un nombre infini de pas qui correspondent à l'autre règle.

À notre avis, un autre point notable dans cette thèse concerne l'utilisation de modèles génériques de réécriture. Nous avons souligné les développements inédites à cet égard, telles que les preuves abstraites de normalisation et de standardisation développés dans le cadre des ARS, et l'extension du modèle fondé sur la notion de *proof term* à des systèmes de réécriture infinitaire. Nous estimons que, en général, le materiel dans cette thèse témoigne de la pertinence de l'utilisation de modèles génériques dans l'étude de calculs dont l'analyse des espaces de réduction est une tâche difficile.

Appendix C

Detailed proofs for the linear substitution calculus

The functions about free and bound variables are defined as expected. The only difference between fv and flv is $\texttt{fv}(x) = \{x\}$, $\texttt{flv}(x) = \varnothing$. We will also refer to the set of bound variables of a term t, $\texttt{bv}(t)$, with the expected definitions. As the contexts are terms, the definitions of fv, flv and bv apply to contexts as well as to terms.

We define the **set of bound labeled variables of a list of substitutions** $\texttt{L} = [x_1^{(\alpha_1)}/t_1] \ldots [x_n^{(\alpha_n)}/t_n]$, as $\texttt{blv}(\texttt{L}) = \{x_i \ / \ x_i^{(\alpha_i)} = x_i^{\alpha_i}\}$, i.e., the subset of variables carrying a label in $\{x_i^{(\alpha_i)}\}$.

We say that a list of substitutions $[x_1^{(\alpha_1)}/t_1] \ldots [x_n^{(\alpha_n)}/t_n]$ is *well-labeled*, if for all i, $t_i \in \mathcal{T}_{\mathcal{WL}}$, and $\texttt{fv}(t_i) \cap \texttt{blv}([x_{i+1}^{(\alpha_{i+1})}/t_{i+1}] \ldots [x_n^{(\alpha_n)}/t_n]) = \varnothing$. We denote the set of **well-labeled lists of substitutions** as $\mathcal{L}_{\mathcal{WL}}$.

We assume α-conversion preserves free variables and well-labeling, i.e.:

Lemma C.0.1. *Let t, u terms such that t and u are α-convertible. Then* $\texttt{fv}(t) = \texttt{fv}(u)$, $\texttt{flv}(t) = \texttt{flv}(u)$, *and* $t \in \mathcal{T}_{\mathcal{WL}}$ *iff* $u \in \mathcal{T}_{\mathcal{WL}}$.

C.1 Invariance of the set $\mathcal{T}_{\mathcal{WL}}$

We prove that well-labeling is invariant w.r.t. reduction and graphical equivalence, i.e., Lem. 4.1.7, stated in page 101. We obtain also Lem. 4.2.7, stated in page 104, as an intermediate result.

Lemma C.1.1. *Let t, u be terms and C a context. Then (i)* $\texttt{fv}(u) \subseteq \texttt{fv}(t)$ *implies* $\texttt{fv}(C[u]) \subseteq \texttt{fv}(C[t])$, *(ii)* $\texttt{flv}(u) \subseteq \texttt{flv}(t)$ *implies* $\texttt{flv}(C[u]) \subseteq \texttt{flv}(C[t])$.

Proof. Straightforward induction on C. □

Lemma C.1.2. *Let* $t\,\texttt{L} \in \mathcal{T}_{\mathcal{WL}}$. *Then* $t \in \mathcal{T}_{\mathcal{WL}}$, $\texttt{L} \in \mathcal{L}_{\mathcal{WL}}$, *and* $\texttt{fv}(t) \cap \texttt{blv}(\texttt{L}) = \varnothing$.

Proof. Easy induction on $|\texttt{L}|$. Notice that $\texttt{L} = \texttt{L}'[x_n^{(\alpha_n)}/t_n]$ implies $t\,\texttt{L} = (t\,\texttt{L}')[x_n^{(\alpha_n)}/t_n]$. □

Lemma C.1.3. *Let* $t \in \mathcal{T}_{\mathcal{WL}}$ *and* $\texttt{L} \in \mathcal{L}_{\mathcal{WL}}$ *such that* $\texttt{fv}(t) \cap \texttt{blv}(\texttt{L}) = \varnothing$. *Then* $t\,\texttt{L} \in \mathcal{T}_{\mathcal{WL}}$.

Proof. Easy induction on $|\mathtt{L}|$. □

Lemma C.1.4. *Let t be a term and C a context. Then:*

(i) $\mathtt{fv}(t)\backslash\mathtt{bv}(C) \subseteq \mathtt{fv}(C[t]) \;\wedge\; \mathtt{fv}(C) \subseteq \mathtt{fv}(C[t]) \subseteq \mathtt{fv}(C) \cup \mathtt{fv}(t)$,

(ii) $\mathtt{flv}(t)\backslash\mathtt{bv}(C) \subseteq \mathtt{flv}(C[t]) \;\wedge\; \mathtt{flv}(C) \subseteq \mathtt{flv}(C[t]) \subseteq \mathtt{flv}(C) \cup \mathtt{flv}(t)$

Proof. Easy induction on C. □

Lemma C.1.5. *Let t, u be terms. Then (i) $t \overset{\alpha}{\mapsto} u$ implies $\mathtt{fv}(t) \supseteq \mathtt{fv}(u)$, and (ii) $t \approx u$ implies $\mathtt{fv}(t) = \mathtt{fv}(u)$.*

Proof. Straightforward case analysis. □

Based on the previous basic properties, we can prove the results leading to Lem. 4.1.7. The first of these results is actually Lem. 4.2.7.

The following proofs use extensively Lem. C.1.4. We will not mention explicitly this fact in each occurrence, to make the text less cumbersome.

Proof of Lem. 4.2.7.
We proceed by induction on C.

- If $C = \Box$ then we conclude immediately.
- Assume $C = \lambda y.C_1$. In this case $C[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $C_1[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ and $y \notin \mathtt{flv}(C_1[\![x^{(\alpha)}]\!])$, so that $y \notin \mathtt{flv}(C_1)$. Moreover $\mathtt{bv}(C) = \mathtt{bv}(C_1) \cup \{y\}$, implying $\mathtt{fv}(t) \cap \mathtt{bv}(C_1) = \varnothing$ and $y \notin \mathtt{fv}(t)$. The former allows to apply IH, obtaining $C_1[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$, while the latter implies $y \notin \mathtt{flv}(t)$ which, along with $y \notin \mathtt{flv}(C_1)$, yield $y \notin \mathtt{flv}(C_1[\![t]\!])$, cfr. Lem. C.1.4. Thus $C[\![t]\!] = \lambda y.C_1[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$.
- The case $C = \lambda y^{\alpha}.C_1$ would contradict $C[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$.
- Assume $C = C_1\, u$, $C_1[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ and $u \in \mathcal{T}_{\mathcal{WL}}$. IH on C_1 suffices to conclude.
- Assume $C = C_1\, u$, $C_1[\![x^{(\alpha)}]\!] = (\lambda y^{\beta}.s)\mathtt{L}$, $(\lambda y.s)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$, and $u \in \mathcal{T}_{\mathcal{WL}}$. There are two cases to consider.

  If $C_1 = (\lambda y^{\beta}.C_2)\mathtt{L}$, so that $C_1[\![x^{(\alpha)}]\!] = (\lambda y^{\beta}.C_2[\![x^{(\alpha)}]\!])\mathtt{L}$, then Lem. C.1.2 implies $\lambda y.C_2[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$, $\mathtt{L} \in \mathcal{L}_{\mathcal{WL}}$, and $\mathtt{fv}(\lambda y.C_2[\![x^{(\alpha)}]\!]) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$. In turn, $\lambda y.C_2[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $C_2[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ and $y \notin \mathtt{flv}(C_2[\![x^{(\alpha)}]\!])$, so that $y \notin \mathtt{flv}(C_2)$. Noticing $\mathtt{bv}(C_2) \cup \{y\} \subseteq \mathtt{bv}(C)$ allows to apply IH to obtain $C_2[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$, and also implies $y \notin \mathtt{flv}(t)$, which along with $y \notin \mathtt{flv}(C_2)$ yields $y \notin \mathtt{flv}(C_2[\![t]\!])$. Therefore $\lambda y.C_2[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$. On the other hand, $\mathtt{fv}(\lambda y.C_2[\![x^{(\alpha)}]\!]) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$ implies $\mathtt{fv}(C_2)\backslash\{y\} \cap \mathtt{blv}(\mathtt{L}) = \varnothing$, and $\mathtt{blv}(\mathtt{L}) \subseteq \mathtt{bv}(C)$ implying $\mathtt{fv}(t) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$, so that $\mathtt{fv}(\lambda y.C_2[\![t]\!]) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$. Hence Lem. C.1.3 yields $(\lambda y.C_2[\![t]\!])\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$. Thus $C[\![t]\!] = (\lambda y^{\beta}.C_2[\![t]\!])\mathtt{L}u \in \mathcal{T}_{\mathcal{WL}}$.

  If $C_1 = (\lambda y^{\beta}.s)\mathtt{L}_1[z^{(\gamma)}/C_2]\mathtt{L}_2$, then Lem. C.1.2 implies $(\lambda y.s)\mathtt{L}_1 \in \mathcal{T}_{\mathcal{WL}}$, $C_2[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$, $\mathtt{fv}(C_2[\![x^{(\alpha)}]\!]) \cap \mathtt{blv}(\mathtt{L}_2) = \varnothing$, $\mathtt{L}_2 \in \mathcal{L}_{\mathcal{WL}}$, and $\mathtt{fv}((\lambda y.s)\mathtt{L}_1) \cap \mathtt{blv}([z^{(\gamma)}/C_2[\![x^{(\alpha)}]\!]]\mathtt{L}_2) = \varnothing$. Observing $\mathtt{bv}(C_2) \subseteq \mathtt{bv}(C)$, we obtain $C_2[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$ from $C_2[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ by IH. Moreover, $\mathtt{blv}(\mathtt{L}_2) \subseteq \mathtt{bv}(C)$, so that $\mathtt{fv}(t) \cap \mathtt{blv}(\mathtt{L}_2) = \varnothing$, and $\mathtt{fv}(C_2[\![x^{(\alpha)}]\!]) \cap \mathtt{blv}(\mathtt{L}_2) = \varnothing$ implies $\mathtt{fv}(C_2) \cap \mathtt{blv}(\mathtt{L}_2) = \varnothing$, so that $\mathtt{fv}(C_2[\![t]\!]) \cap \mathtt{blv}(\mathtt{L}_2) = \varnothing$. Therefore $[z^{(\gamma)}/C_2[\![t]\!]]\mathtt{L}_2 \in \mathcal{L}_{\mathcal{WL}}$. Furthermore, $\mathtt{fv}((\lambda y.s)\mathtt{L}_1) \cap \mathtt{blv}([z^{(\gamma)}/C_2[\![t]\!]]\mathtt{L}_2) = \mathtt{fv}((\lambda y.s)\mathtt{L}_1) \cap \mathtt{blv}([z^{(\gamma)}/C_2[\![x^{(\alpha)}]\!]]\mathtt{L}_2) = \varnothing$. Hence Lem. C.1.3 yields $(\lambda y.s)\mathtt{L}_1[z^{(\gamma)}/C_2[\![t]\!]]\mathtt{L}_2 \in \mathcal{T}_{\mathcal{WL}}$. Thus $C[\![t]\!] = (\lambda y^{\beta}.s)\mathtt{L}_1[z^{(\gamma)}/C_2[\![t]\!]]\mathtt{L}_2\, u \in \mathcal{T}_{\mathcal{WL}}$.

- Assume $C = u\,C_2$. Then $C[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $C_2[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$, and either $u \in \mathcal{T}_{\mathcal{WL}}$ or $u = (\lambda y^\beta.s)\mathtt{L}$ and $(\lambda y.s)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$. In both cases, IH on C_2 suffices to conclude.
- If $C = C_1[y/u]$, then IH on C_1 suffices to conclude.
- If $C = C_1[y^\beta/u]$, then $C[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $C_1[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$, $u \in \mathcal{T}_{\mathcal{WL}}$, and $y \notin \mathtt{fv}(C_1[\![x^{(\alpha)}]\!])$. Observing $\mathtt{bv}(C_1) \subseteq \mathtt{bv}(C)$, we obtain $C_1[\![t]\!] \in \mathcal{T}_{\mathcal{WL}}$ from $C_1[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ by IH. Moreover, $y \notin \mathtt{fv}(C_1[\![x^{(\alpha)}]\!])$ implies $y \notin \mathtt{fv}(C_1)$, and $y \in \mathtt{bv}(C)$ implies $y \notin \mathtt{fv}(t)$, so that $y \notin \mathtt{fv}(C_1[\![t]\!])$. Thus $C[\![t]\!] = C_1[\![t]\!][y^\beta/u] \in \mathcal{T}_{\mathcal{WL}}$.
- If $C = u[y^{(\beta)}/C_2]$, then $C[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $u \in \mathcal{T}_{\mathcal{WL}}$, $y \notin \mathtt{fv}(u)$ if $y^{(\beta)} = y^\beta$, and $C_2[\![x^{(\alpha)}]\!] \in \mathcal{T}_{\mathcal{WL}}$. IH on C_2 suffices to conclude.

□

Lemma C.1.6. *Let $t \overset{\alpha}{\mapsto} u$ such that $t \in \mathcal{T}_{\mathcal{WL}}$. Then $u \in \mathcal{T}_{\mathcal{WL}}$.*

Proof. There are three cases to consider.

If $t \overset{\alpha}{\mapsto}_{\mathtt{db}} u$, i.e. $t = (\lambda x^\alpha.s_1)\mathtt{L}s_2$ and $u = s_1[x/s_2]\mathtt{L}$, then $t \in \mathcal{T}_{\mathcal{WL}}$ implies $(\lambda x.s_1)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$, so that Lem. C.1.2 implies $\lambda x.s_1 \in \mathcal{T}_{\mathcal{WL}}$, $\mathtt{L} \in \mathcal{L}_{\mathcal{WL}}$, and $\mathtt{fv}(\lambda x.s_1) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$, and also $s_2 \in \mathcal{T}_{\mathcal{WL}}$. In turn $\lambda x.s_1 \in \mathcal{T}_{\mathcal{WL}}$ implies $s_1 \in \mathcal{T}_{\mathcal{WL}}$, and therefore $s_1[x/s_2] \in \mathcal{T}_{\mathcal{WL}}$. On the other hand, variable convention implies $\mathtt{fv}(s_2) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$, which along with $\mathtt{fv}(\lambda x.s_1) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$ yield $\mathtt{fv}(s_1[x/s_2]) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$. Hence Lem. C.1.3 implies $u = s_1[x/s_2]\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$.

If $t \overset{\alpha}{\mapsto}_{\mathtt{ls}} u$, i.e. $t = C[\![x^\alpha]\!][x/s]$ and $u = C[\![s]\!][x/s]$, then $C[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$ and $s \in \mathcal{T}_{\mathcal{WL}}$. Moreover, variable convention implies $\mathtt{fv}(s) \cap \mathtt{bv}(C) = \varnothing$, so that Lem. 4.2.7 yields $C[\![s]\!] \in \mathcal{T}_{\mathcal{WL}}$, which suffices to conclude.

If $t = s_1[x^\alpha/s_2]$ and $u = s_1$, then we conclude immediately. □

Lemma C.1.7. *Let $t \approx u$. Then $t \in \mathcal{T}_{\mathcal{WL}}$ iff $u \in \mathcal{T}_{\mathcal{WL}}$.*

Proof. There are three cases to consider.

Assume $t \approx_{\mathtt{CS}} u$, i.e. $t = s_1[x^{(\alpha)}/s_2][y^{(\beta)}/s_3]$, $u = s_1[y^{(\beta)}/s_3][x^{(\alpha)}/s_2]$, $x \notin \mathtt{fv}(s_3)$ and $y \notin \mathtt{fv}(s_2)$, so that variable convention implies $x \neq y$.

Then $t \in \mathcal{T}_{\mathcal{WL}}$ implies $s_1, s_2, s_3 \in \mathcal{T}_{\mathcal{WL}}$, $x \notin \mathtt{fv}(s_1)$ if $x^{(\alpha)} = x^\alpha$, and $y \notin \mathtt{fv}(s_1)$ if $y^{(\beta)} = y^\beta$. Then it is immediate to obtain $s_1[y^{(\beta)}/s_3] \in \mathcal{T}_{\mathcal{WL}}$, and subsequently $u \in \mathcal{T}_{\mathcal{WL}}$.

On the other hand, if $u \in \mathcal{T}_{\mathcal{WL}}$, then we obtain $t \in \mathcal{T}_{\mathcal{WL}}$ analogously.

Assume $t \approx_{\sigma_1} u$, i.e. $t = (\lambda y^{(\beta)}.s_1)[x^{(\alpha)}/s_2]$, $u = \lambda y^{(\beta)}.s_1[x^{(\alpha)}/s_2]$, and $y \notin \mathtt{fv}(s_2)$. In this case, variable convention implies $x \neq y$.

Then $t \in \mathcal{T}_{\mathcal{WL}}$ implies $y^{(\beta)} = y$, and therefore $s_1 \in \mathcal{T}_{\mathcal{WL}}$, $y \notin \mathtt{flv}(s_1)$, $s_2 \in \mathcal{T}_{\mathcal{WL}}$, and $x \notin \mathtt{fv}(\lambda y.s_1)$ if $x^{(\alpha)} = x^\alpha$. Moreover, $x \neq y$ implies that $x \notin \mathtt{fv}(s_1)$ if $x^{(\alpha)} = x^\alpha$. Hence, it is immediate to obtain $s_1[x^{(\alpha)}/s_2] \in \mathcal{T}_{\mathcal{WL}}$, and subsequently $u \in \mathcal{T}_{\mathcal{WL}}$, recalling that $y \notin \mathtt{fv}(s_2) \supseteq \mathtt{flv}(s_2)$.

On the other hand, $u \in \mathcal{T}_{\mathcal{WL}}$ implies $y^{(\beta)} = y$, $s_1[x^{(\alpha)}/s_2] \in \mathcal{T}_{\mathcal{WL}}$, so that $s_1, s_2 \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \mathtt{fv}(s_1)$ if $x^{(\alpha)} = x^\alpha$, and also $y \notin \mathtt{flv}(s_1)$, since $x \neq y$. Hence we obtain immediately $\lambda y.s_1 \in \mathcal{T}_{\mathcal{WL}}$, and subsequently $t \in \mathcal{T}_{\mathcal{WL}}$.

Assume $t \approx_{\sigma_2} u$, i.e. $t = (s_1 s_2)[x^{(\alpha)}/s_3]$, $u = s_1[x^{(\alpha)}/s_3]s_2$, and $x \notin \mathtt{fv}(s_2)$.

Then $t \in \mathcal{T}_{\mathcal{WL}}$ implies $s_1 s_2 \in \mathcal{T}_{\mathcal{WL}}$, $s_3 \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \mathtt{fv}(s_1)$ if $x^{(\alpha)} = x^\alpha$. There are two cases to consider for $s_1 s_2 \in \mathcal{T}_{\mathcal{WL}}$. If $s_1, s_2 \in \mathcal{T}_{\mathcal{WL}}$, then we obtain immediately that $s_1[x^{(\alpha)}/s_3]$, and subsequently u, are in $\mathcal{T}_{\mathcal{WL}}$. If $s_1 = (\lambda y^\beta.s_1')\mathtt{L}$, and $(\lambda y.s_1')\mathtt{L}, s_2 \in \mathcal{T}_{\mathcal{WL}}$, then $\mathtt{fv}((\lambda y.s_1')\mathtt{L}) = \mathtt{fv}(s_1)$, implying that $x \notin \mathtt{fv}((\lambda y.s_1')\mathtt{L})$ if $x^{(\alpha)} = x^\alpha$, and therefore $(\lambda y.s_1')\mathtt{L}[x^{(\alpha)}/s_3] \in \mathcal{T}_{\mathcal{WL}}$. Hence $u = (\lambda y^\beta.s_1')\mathtt{L}[x^{(\alpha)}/s_3]s_2 \in \mathcal{T}_{\mathcal{WL}}$.

On the other hand, there are two cases to consider for $u \in \mathcal{T}_{\mathcal{WL}}$. If $s_1 = (\lambda y^\beta.s_1')\mathtt{L}$, and $(\lambda y.s_1')\mathtt{L}[x^{(\alpha)}/s_3], s_2 \in \mathcal{T}_{\mathcal{WL}}$, then $(\lambda y.s_1')\mathtt{L}, s_3 \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \mathtt{fv}((\lambda y.s_1')\mathtt{L})$ if $x^{(\alpha)} = x^\alpha$. Therefore, $s_1 s_2 \in \mathcal{T}_{\mathcal{WL}}$. Moreover, $\mathtt{fv}(s_1) = \mathtt{fv}((\lambda y.s_2')\mathtt{L})$, implying $x \notin \mathtt{fv}(s_1)$ if $x^{(\alpha)} = x^\alpha$, which along with $x \notin \mathtt{fv}(s_2)$ and $s_3 \in \mathcal{T}_{\mathcal{WL}}$, imply $t \in \mathcal{T}_{\mathcal{WL}}$. □

Lemma C.1.8. *Let $t \in \mathcal{T}_{\mathcal{WL}}$ and u a term, so that $t \xrightarrow{\alpha} u$ or $t \overset{1}{\sim} u$. Then $u \in \mathcal{T}_{\mathcal{WL}}$.*

Proof. The hypothesis implies $t = C[t']$, $u = C[u']$, and $t' \overset{\alpha}{\mapsto} u'$, $t' \approx u'$, $u' \approx t'$, or u' is the result of applying one step of α-conversion from t' or vice versa. We proceed by induction on C.

- If $C = \Box$, we conclude immediately by Lem. C.1.6, Lem. C.1.7 or Lem. C.0.1.
- If $C = \lambda x.C_1$, then $t \in \mathcal{T}_{\mathcal{WL}}$ implies $C_1[t'] \in \mathcal{T}_{\mathcal{WL}}$, so that IH yields $C_1[u'] \in \mathcal{T}_{\mathcal{WL}}$, and $x \notin \mathtt{flv}(C_1[t'])$. As moreover variable convention implies $x \notin \mathtt{bv}(C_1)$, we can obtain $x \notin \mathtt{flv}(t')$, and therefore $x \notin \mathtt{flv}(u')$ by Lem. C.1.5 or Lem. C.0.1. Observe that $x \notin \mathtt{flv}(C_1[t'])$ implies also $x \notin \mathtt{flv}(C_1)$, which along with $x \notin \mathtt{flv}(u')$ yields $x \notin \mathtt{flv}(C_1[u'])$. Thus $C[u'] = \lambda x.C_1[u'] \in \mathcal{T}_{\mathcal{WL}}$.
- The case $C = \lambda x^\alpha.C_1$ would contradict $C[t] \in \mathcal{T}_{\mathcal{WL}}$.
- If $C = C_1\, s$ and $C_1[t'], s \in \mathcal{T}_{\mathcal{WL}}$, then IH suffices to conclude.
- Assume $C = C_1\, s$, $C_1[t'] = (\lambda x^\alpha.s_1)\mathtt{L}$, and $(\lambda x.s_1)\mathtt{L}, s \in \mathcal{T}_{\mathcal{WL}}$. There are several cases to analyse.
  - $C_1 = \Box$, $t' = (\lambda x^\alpha.s_1)\mathtt{L}$. A case analysis on $\overset{\alpha}{\mapsto}$, $\approx$ and α-conversion implies that $u' = (\lambda y^\alpha.s_1')\mathtt{L}'$ where $(\lambda x.s_1)\mathtt{L} \overset{\alpha}{\mapsto} (\lambda y.s_1')\mathtt{L}'$, $(\lambda x.s_1)\mathtt{L} \approx (\lambda y.s_1')\mathtt{L}'$ or vice versa, or $(\lambda x.s_1)\mathtt{L}$ and $(\lambda y.s_1')\mathtt{L}'$ are one application of α-conversion away. Therefore Lem. C.1.6, Lem. C.1.7 or Lem. C.0.1 yields $(\lambda y.s_1')\mathtt{L}' \in \mathcal{T}_{\mathcal{WL}}$, hence $C[u'] = (\lambda y^\alpha.s_1')\mathtt{L}'s \in \mathcal{T}_{\mathcal{WL}}$.
  - $C_1 = \Box\mathtt{L}$ and $t' = \lambda x^\alpha.s_1$. Observe that $(\lambda x.s_1)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$ implies $\lambda x.s_1 \in \mathcal{T}_{\mathcal{WL}}$, $\mathtt{L} \in \mathcal{L}_{\mathcal{WL}}$ and $\mathtt{fv}(\lambda x.s_1) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$, by Lem. C.1.2. Case analysis yields $u' = (\lambda y^\alpha.s_1')\mathtt{L}'$, and $(\lambda y.s_1')\mathtt{L}' \approx_{\sigma_1} \lambda x.s_1$, or $\lambda x.s_1$ and $(\lambda y.s_1')\mathtt{L}'$ are one application of α-conversion away (in the latter case, $\mathtt{L}'$ is the empty list). Therefore, Lem. C.1.7 or Lem. C.0.1 implies $(\lambda y.s_1')\mathtt{L}' \in \mathcal{T}_{\mathcal{WL}}$. In turn, Lem. C.1.5 or Lem. C.0.1 implies $\mathtt{fv}((\lambda y.s_1')\mathtt{L}') = \mathtt{fv}(\lambda x.s_1)$. Hence, Lem. C.1.3 implies that $(\lambda y.s_1')\mathtt{L}'\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$. Thus, $C[u'] = (\lambda y^\alpha.s_1')\mathtt{L}'\mathtt{L}s \in \mathcal{T}_{\mathcal{WL}}$.
  - $C_1 = (\lambda x^\alpha.C_2)\mathtt{L}$, so that $(\lambda x.C_2[t'])\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$. Lem. C.1.2 implies $\lambda x.C_2[t'] \in \mathcal{T}_{\mathcal{WL}}$, $\mathtt{L} \in \mathcal{L}_{\mathcal{WL}}$, and $\mathtt{fv}(\lambda x.C_2[t']) \cap \mathtt{blv}(\mathtt{L}) = (\mathtt{fv}(C_2[t'])\backslash\{x\}) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$. In turn, $\lambda x.C_2[t'] \in \mathcal{T}_{\mathcal{WL}}$ implies $C_2[t'] \in \mathcal{T}_{\mathcal{WL}}$, so that IH yields $C_2[u'] \in \mathcal{T}_{\mathcal{WL}}$, and also $x \notin \mathtt{flv}(C_2[t'])$. On the other hand, Lem. C.1.5 and Lem. C.1.1 imply $\mathtt{flv}(C_2[u']) \subseteq \mathtt{flv}(C_2[t'])$ and $\mathtt{fv}(C_2[u']) \subseteq \mathtt{fv}(C_2[t'])$, so that particularly $x \notin \mathtt{flv}(C_2[u'])$. Hence $C_2[u'] \in \mathcal{T}_{\mathcal{WL}}$ implies $\lambda x.C_2[u'] \in \mathcal{T}_{\mathcal{WL}}$. Notice that $\mathtt{fv}(C_2[u']) \subseteq \mathtt{fv}(C_2[t'])$ implies $(\mathtt{fv}(C_2[u'])\backslash\{x\}) \cap \mathtt{blv}(\mathtt{L}) = \varnothing$. Consequently, Lem. C.1.3 yields $(\lambda x.C_2[u'])\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$. Thus $C[u'] = (\lambda x^\alpha.C_2[u'])\mathtt{L}s \in \mathcal{T}_{\mathcal{WL}}$.

- $C_1 = (\lambda x^\alpha.s_1)\mathtt{L}_1[y^{(\beta)}/C_2]\mathtt{L}_2$, so that $(\lambda x.s_1)\mathtt{L}_1[y^{(\beta)}/C_2[t']]\mathtt{L}_2 \in \mathcal{T}_{\mathcal{WL}}$. In this case, Lem. C.1.2 implies $(\lambda x.s_1)\mathtt{L}_1 \in \mathcal{T}_{\mathcal{WL}}$, $[y^{(\beta)}/C_2[t']]\mathtt{L}_2 \in \mathcal{L}_{\mathcal{WL}}$, and $\mathtt{fv}((\lambda x.s_1)\mathtt{L}_1) \cap \mathtt{blv}([y^{(\beta)}/C_2[t']]\mathtt{L}_2) = \emptyset$. In turn, $[y^{(\beta)}/C_2[t']]\mathtt{L}_2 \in \mathcal{L}_{\mathcal{WL}}$ implies $C_2[t'] \in \mathcal{T}_{\mathcal{WL}}$ and $\mathtt{fv}(C_2[t']) \cap \mathtt{blv}(\mathtt{L}_2) = \emptyset$. Therefore, IH yields $C_2[u'] \in \mathcal{T}_{\mathcal{WL}}$. Moreover, $\mathtt{fv}(C_2[u']) \subseteq \mathtt{fv}(C_2[t'])$ by Lem. C.1.1, implying $\mathtt{fv}(C_2[u']) \cap \mathtt{blv}(\mathtt{L}_2) = \emptyset$. Hence $[y^{(\beta)}/C_2[u']]\mathtt{L}_2 \in \mathcal{L}_{\mathcal{WL}}$, so that Lem. C.1.3 implies $(\lambda x.s_1)\mathtt{L}_1[y^{(\beta)}/C_2[u']]\mathtt{L}_2 \in \mathcal{T}_{\mathcal{WL}}$. Thus $C[u'] = (\lambda x^\alpha.s_1)\mathtt{L}_1[y^{(\beta)}/C_2[u']]\mathtt{L}_2 s \in \mathcal{T}_{\mathcal{WL}}$.

- If $C = sC_2$, then $C_2[t'] \in \mathcal{T}_{\mathcal{WL}}$, and either $s \in \mathcal{T}_{\mathcal{WL}}$ or $s = (\lambda x^\alpha.s_1)\mathtt{L}$ and $(\lambda x.s_1)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$. In both cases, IH suffices to conclude.
- If $C = C_1[x/s]$, then $C_1[t'], s \in \mathcal{T}_{\mathcal{WL}}$, hence IH suffices to conclude.
- If $C = C_1[x^\alpha/s]$, then $C_1[t'], s \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \mathtt{fv}(C_1[t'])$. IH implies $C_1[u'] \in \mathcal{T}_{\mathcal{WL}}$. On the other hand, Lem. C.1.5 and Lem. C.1.1 imply $\mathtt{fv}(C_1[u']) \subseteq \mathtt{fv}(C_1[t'])$, therefore $x \notin \mathtt{fv}(C_1[u'])$. Hence $C[u'] = C_1[u'][x^\alpha/s] \in \mathcal{T}_{\mathcal{WL}}$.
- If $C = s[x^{(\alpha)}/C_1]$, then $s, C_1[t'] \in \mathcal{T}_{\mathcal{WL}}$ and $x \notin \mathtt{fv}(s)$ if $x^{(\alpha)} = x^\alpha$. Therefore, IH suffices to conclude.

□

Proof of Lem. 4.1.7.
Given $t \in \mathcal{T}_{\mathcal{WL}}$ and $t \xrightarrow{\alpha} u$ or $t \sim u$, we must show $u \in \mathcal{T}_{\mathcal{WL}}$. If $t \xrightarrow{\alpha} u$, then Lem. C.1.8 allows to conclude immediately. If $t \sim u$, then a straightforward induction on the sequence $t = t_0 \overset{1}{\sim} t_1 \overset{1}{\sim} \ldots \overset{1}{\sim} t_n = u$, based on Lem. C.1.8, is enough to conclude. □

C.2 Finite developments

We give the proof of some lemmas stated, and used, in Section 4.2.1.

Proof of Lem. 4.2.9.
We recall the statement.
Let $C[\![x^\alpha]\!], u \in \mathcal{T}_{\mathcal{WL}}$ and a variable y such that $x \neq y$, $\mathtt{fv}(u) \cap \mathtt{bv}(C) = \emptyset$, and $x, y \notin \mathtt{fv}(u)$. Then: *(i)* $\mathtt{LM}_x(C[\![x^\alpha]\!]) > \mathtt{LM}_x(C[\![u]\!])$, and *(ii)* $\mathtt{LM}_y(C[\![x^\alpha]\!]) = \mathtt{LM}_y(C[\![u]\!])$.

The proof proceeds by induction on $|C|$.

- If $C = \Box$, then $\mathtt{LM}_x(x^\alpha) = 1 > 0 = \mathtt{LM}_x(u)$, and $\mathtt{LM}_y(x^\alpha) = 0 = \mathtt{LM}_y(u)$.
- If $C = \lambda z.C_1$, then $C[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $C_1[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$, therefore IH suffices to conclude. Notice that $z = x$ would contradict $\lambda z.C_1[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$.
- If $C = C_1 s$ and $C_1[\![x^\alpha]\!], s \in \mathcal{T}_{\mathcal{WL}}$, then a straightforward inductive argument suffices to conclude.
- Assume $C = C_1 s$, $C_1[\![x^\alpha]\!] = (\lambda z^\beta.s_1)\mathtt{L}$, $(\lambda z.s_1)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$, $s \in \mathcal{T}_{\mathcal{WL}}$. In this case, $C_1[\![x^\alpha]\!] = (\lambda z^\beta.s_1)\mathtt{L}$ implies $C_1 = (\lambda z^\beta.C_2)\mathtt{L}$ or $C_1 = (\lambda z^\beta.s_1)\mathtt{L}_1[x_i^{(\gamma)}/C_2]\mathtt{L}_2$. Let C_1' be the result of replacing z^β by z in C_1. Then $|C_1'| = |C_1| < |C|$ and $C_1'[\![x^\alpha]\!] = (\lambda z.s_1)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$, so that IH yields
$$\mathtt{LM}_x(C_1'[\![x^\alpha]\!]) > \mathtt{LM}_x(C_1'[\![u]\!]) \text{ and } \mathtt{LM}_y(C_1'[\![x^\alpha]\!]) = \mathtt{LM}_y(C_1'[\![u]\!])$$
We conclude by observing that $\mathtt{LM}_x(C[\![x^\alpha]\!]) = \mathtt{LM}_x(C_1'[\![x^\alpha]\!]) + \mathtt{LM}_x(s)$, analogously for $\mathtt{LM}_x(C[\![u]\!])$, and analogously again for $\mathtt{LM}_y$.

- Assume $C = sC_2$, then $C_2[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$. Let us define s' as follows: $s' = s$ if $s \in \mathcal{T}_{\mathcal{WL}}$, or $s' = (\lambda z.s_1)\mathtt{L}$ if $s = (\lambda z^\beta.s_1)\mathtt{L}$. Then $\mathtt{LM}_x(C[\![x^\alpha]\!]) = \mathtt{LM}_x(s') + \mathtt{LM}_x(C_2[\![x^\alpha]\!])$, an analogously for $\mathtt{LM}_x(C[\![u]\!])$ and for $\mathtt{LM}_y$. Hence IH suffices to conclude.

- If $C = C_1[z/s]$, so that $C_1[\![x^\alpha]\!], s \in \mathcal{T}_{\mathcal{WL}}$, then the occurrences of z in $C[t]$ are bound for any t, implying $x \neq z$ and $z \notin \mathtt{fv}(u)$. IH implies $\mathtt{LM}_x(C_1[\![x^\alpha]\!]) > \mathtt{LM}_x(C_1[\![u]\!])$, $\mathtt{LM}_y(C_1[\![x^\alpha]\!]) = \mathtt{LM}_y(C_1[\![u]\!])$, and also $\mathtt{LM}_z(C_1[\![x^\alpha]\!]) = \mathtt{LM}_z(C_1[\![u]\!])$. Therefore, for *(i)* we have
$$\begin{array}{rcl} \mathtt{LM}_x(C[\![x^\alpha]\!][z/s]) & = & \mathtt{LM}_x(C_1[\![x^\alpha]\!]) + \mathtt{LM}_x(s) + \mathtt{LM}_z(C_1[\![x^\alpha]\!]) \cdot \mathtt{LM}_x(s) \\ & > & \mathtt{LM}_x(C_1[\![u]\!]) + \mathtt{LM}_x(s) + \mathtt{LM}_z(C_1[\![u]\!]) \cdot \mathtt{LM}_x(s) \\ & = & \mathtt{LM}_x(C[\![u]\!]) \end{array}$$
For *(ii)*, if $z = y$ then immediately $\mathtt{LM}_y(C[\![x^\alpha]\!]) = \mathtt{LM}_y(C[\![u]\!]) = 0$, otherwise an analysis similar to that used for *(i)* applies.

- If $C = s[z/C_2]$, so that $s, C_2[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$, variable convention implies $x \neq z$. IH implies $\mathtt{LM}_x(C_2[\![x^\alpha]\!]) > \mathtt{LM}_x(C_2[\![u]\!])$ and $\mathtt{LM}_y(C_2[\![x^\alpha]\!]) = \mathtt{LM}_y(C_2[\![u]\!])$. For *(i)* we have
$$\begin{array}{rcl} \mathtt{LM}_x(C[\![x^\alpha]\!][z/s]) & = & \mathtt{LM}_x(s) + \mathtt{LM}_x(C_2[\![x^\alpha]\!]) + \mathtt{LM}_z(s) \cdot \mathtt{LM}_x(C_2[\![x^\alpha]\!]) \\ & > & \mathtt{LM}_x(s) + \mathtt{LM}_x(C_2[\![u]\!]) + \mathtt{LM}_z(s) \cdot \mathtt{LM}_x(C_2[\![u]\!]) \\ & = & \mathtt{LM}_x(C[\![u]\!]) \end{array}$$
For *(ii)*, if $z = y$ then immediately $\mathtt{LM}_y(C[\![x^\alpha]\!]) = \mathtt{LM}_y(C[\![u]\!]) = 0$, otherwise an analysis similar to that used for *(i)* applies.

- If $C = C_1[z^\beta/s]$, or $C = s[z^\beta/C_2]$, then we obtain $x \neq z$ like in the previous cases. A simple argument based on IH suffices to conclude, except for *(ii)* if $z \neq y$; in this case immediately $\mathtt{LM}_y(C[\![x^\alpha]\!]) = \mathtt{LM}_y(C[\![u]\!]) = 0$.

□

Proof of Lem. 4.2.10.
We recall the statement.
Let $C[\![y^\gamma]\!] \in \mathcal{T}_{\mathcal{WL}}$, $u \in \mathcal{T}_{\mathcal{WL}}$ and x variable, such that $x \neq y$, $y \notin \mathtt{fv}(u)$ and $x \notin \mathtt{bv}(C)$. Then $\mathtt{LM}_x(C[\![y^\gamma]\!]) + \mathtt{LM}_y(C[\![y^\gamma]\!]) \cdot \mathtt{LM}_x(u) = \mathtt{LM}_x(C[\![u]\!]) + \mathtt{LM}_y(C[\![u]\!]) \cdot \mathtt{LM}_x(u)$.

The proof proceeds by induction on $|C|$.

- If $C = \Box$, then $\mathtt{LM}_x(C[\![y^\gamma]\!]) + \mathtt{LM}_y(C[\![y^\gamma]\!]) \cdot \mathtt{LM}_x(u) = \mathtt{LM}_x(C[\![u]\!]) + \mathtt{LM}_y(C[\![u]\!]) \cdot \mathtt{LM}_x(u) = \mathtt{LM}_x(u)$.

- If $C = \lambda z.C_1$, then $C[\![y^\gamma]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $z \neq y$. A straightforward inductive argument suffices to conclude.

- If $C = C_1 s$ and $C_1[\![y^\gamma]\!], s \in \mathcal{T}_{\mathcal{WL}}$, then we have
$$\begin{array}{l} \mathtt{LM}_x(C[\![y^\gamma]\!]) + \mathtt{LM}_y(C[\![y^\gamma]\!]) \cdot \mathtt{LM}_x(u) \\ \quad = \mathtt{LM}_x(C_1[\![y^\gamma]\!]) + \mathtt{LM}_x(s) + (\mathtt{LM}_y(C_1[\![y^\gamma]\!]) + \mathtt{LM}_y(s)) \cdot \mathtt{LM}_x(u) \\ \quad = \mathtt{LM}_x(C_1[\![y^\gamma]\!]) + \mathtt{LM}_y(C_1[\![y^\gamma]\!]) \cdot \mathtt{LM}_x(u) + \mathtt{LM}_x(s) + \mathtt{LM}_y(s) \cdot \mathtt{LM}_x(u) \end{array}$$
Analogously we obtain
$$\begin{array}{l} \mathtt{LM}_x(C[\![u]\!]) + \mathtt{LM}_y(C[\![u]\!]) \cdot \mathtt{LM}_x(u) \\ \quad = \mathtt{LM}_x(C_1[\![u]\!]) + \mathtt{LM}_y(C_1[\![u]\!]) \cdot \mathtt{LM}_x(u) + \mathtt{LM}_x(s) + \mathtt{LM}_y(s) \cdot \mathtt{LM}_x(u) \end{array}$$
Hence, IH suffices to conclude.

- Assume $C = C_1 s$, $C_1[\![y^\gamma]\!] = (\lambda z^\beta.s_1)\mathtt{L}$, $(\lambda z.s_1)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$, $s \in \mathcal{T}_{\mathcal{WL}}$. Let us define C_1' as in the analogous case in the proof of Lem. 4.2.9, so that $C_1'[\![y^\gamma]\!] = (\lambda z.s_1)\mathtt{L}$. Observe that $\mathtt{LM}_x(C[\![y^\gamma]\!]) = \mathtt{LM}_x(C_1'[\![y^\gamma]\!]) + \mathtt{LM}_x(s)$, analogously for $\mathtt{LM}_x(C[\![u]\!])$, and also for $\mathtt{LM}_y$. Moreover, IH applies to C_1'. Hence, an argument similar to that used in the previous case applies.
- If $C = sC_2$, so that $C_2[\![y^\gamma]\!] \in \mathcal{T}_{\mathcal{WL}}$, then let us define s' as in the analogous case in the proof of Lem. 4.2.9, so that $\mathtt{LM}_x(C[\![y^\gamma]\!]) = \mathtt{LM}_x(s') + \mathtt{LM}_x(C_2[\![y^\gamma]\!])$ and analogously for $\mathtt{LM}_y$. An argument similar to that of the third case in this proof, considering s', applies.
- If $C = C_1[z/s]$, so that $C_1[\![y^\gamma]\!], s \in \mathcal{T}_{\mathcal{WL}}$, then the occurrences of z in $C[t]$ are bound for any t, implying $y \neq z$ and $z \notin \mathtt{fv}(u)$. Also, $x \notin \mathtt{bv}(C)$ implies $x \neq z$. Let us call $\mathsf{s_x} = \mathtt{LM}_x(s)$, $\mathsf{s_y} = \mathtt{LM}_y(s)$ and $\mathsf{u_x} = \mathtt{LM}_x(u)$. Then

$$\begin{aligned}&\mathtt{LM}_x(C[\![y^\gamma]\!]) + \mathtt{LM}_y(C[\![y^\gamma]\!]) \cdot \mathtt{LM}_x(u)\\ &\quad = \mathtt{LM}_x(C_1[\![y^\gamma]\!]) + \mathsf{s_x} + \mathtt{LM}_z(C_1[\![y^\gamma]\!]) \cdot \mathsf{s_x} + (\mathtt{LM}_y(C_1[\![y^\gamma]\!]) + \mathsf{s_y} + \mathtt{LM}_z(C_1[\![y^\gamma]\!]) \cdot \mathsf{s_y}) \cdot \mathsf{u_x}\\ &\quad = \mathtt{LM}_x(C_1[\![y^\gamma]\!]) + \mathtt{LM}_y(C_1[\![y^\gamma]\!]) \cdot \mathsf{u_x} + \mathtt{LM}_z(C_1[\![y^\gamma]\!]) \cdot (\mathsf{s_x} + \mathsf{s_y} \cdot \mathsf{u_x}) + \mathsf{s_x} + \mathsf{s_y} \cdot \mathsf{u_x}\end{aligned}$$

  Analogously we obtain

$$\begin{aligned}&\mathtt{LM}_x(C[\![u]\!]) + \mathtt{LM}_y(C[\![u]\!]) \cdot \mathtt{LM}_x(u)\\ &\quad = \mathtt{LM}_x(C_1[\![u]\!]) + \mathtt{LM}_y(C_1[\![u]\!]) \cdot \mathsf{u_x} + \mathtt{LM}_z(C_1[\![u]\!]) \cdot (\mathsf{s_x} + \mathsf{s_y} \cdot \mathsf{u_x}) + \mathsf{s_x} + \mathsf{s_y} \cdot \mathsf{u_x}\end{aligned}$$

  Moreover, Lem. 4.2.9 implies $\mathtt{LM}_z(C_1[\![y^\gamma]\!]) = \mathtt{LM}_z(C_1[\![u]\!])$. Hence IH suffices to conclude.
- If $C = s[z/C_2]$, so that $s, C_2[\![y^\gamma]\!] \in \mathcal{T}_{\mathcal{WL}}$, variable convention implies $y \neq z$, and $x \notin \mathtt{bv}(C)$ implies $x \neq z$. We add $\mathsf{s_z} = \mathtt{LM}_z(s)$ to the acronyms used in the previous case. Then

$$\begin{aligned}&\mathtt{LM}_x(C[\![y^\gamma]\!]) + \mathtt{LM}_y(C[\![y^\gamma]\!]) \cdot \mathtt{LM}_x(u)\\ &\quad = \mathsf{s_x} + \mathtt{LM}_x(C_2[\![y^\gamma]\!]) + \mathsf{s_z} \cdot \mathtt{LM}_x(C_2[\![y^\gamma]\!]) + (\mathsf{s_y} + \mathtt{LM}_y(C_2[\![y^\gamma]\!]) + \mathsf{s_z} \cdot \mathtt{LM}_y(C_2[\![y^\gamma]\!])) \cdot \mathsf{u_x}\\ &\quad = \mathtt{LM}_x(C_2[\![y^\gamma]\!]) + \mathtt{LM}_y(C_2[\![y^\gamma]\!]) \cdot \mathsf{u_x} + \mathsf{s_z} \cdot \mathtt{LM}_x(C_2[\![y^\gamma]\!]) + \mathsf{s_z} \cdot \mathtt{LM}_y(C_2[\![y^\gamma]\!]) \cdot \mathsf{u_x} + \mathsf{s_x} + \mathsf{s_y} \cdot \mathsf{u_x}\\ &\quad = \mathtt{LM}_x(C_2[\![y^\gamma]\!]) + \mathtt{LM}_y(C_2[\![y^\gamma]\!]) \cdot \mathsf{u_x} + \mathsf{s_z} \cdot (\mathtt{LM}_x(C_2[\![y^\gamma]\!]) + \mathtt{LM}_y(C_2[\![y^\gamma]\!]) \cdot \mathsf{u_x}) + \mathsf{s_x} + \mathsf{s_y} \cdot \mathsf{u_x}\end{aligned}$$

  Analogously we obtain

$$\begin{aligned}&\mathtt{LM}_x(C[\![u]\!]) + \mathtt{LM}_y(C[\![u]\!]) \cdot \mathtt{LM}_x(u)\\ &\quad = \mathtt{LM}_x(C_2[\![u]\!]) + \mathtt{LM}_y(C_2[\![u]\!]) \cdot \mathsf{u_x} + \mathsf{s_z} \cdot (\mathtt{LM}_x(C_2[\![u]\!]) + \mathtt{LM}_y(C_2[\![u]\!]) \cdot \mathsf{u_x}) + \mathsf{s_x} + \mathsf{s_y} \cdot \mathsf{u_x}\end{aligned}$$

  Hence IH suffices to conclude.
- If $C = C_1[z^\beta/s]$, or $C = s[z^\beta/C_2]$, then we obtain $y \neq z$ and $x \neq z$ like in the previous cases. An inductive argument like in the third case allows to conclude.

□

Proof of Lem. 4.2.11.
We recall the statement.
Let $C[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$ and $u \in \mathcal{T}_{\mathcal{WL}}$ such that $x \notin \mathtt{fv}(u)$. Then $\mathtt{PLR}(C[\![x^\alpha]\!]) + \mathtt{LM}_x(C[\![x^\alpha]\!]) \cdot \mathtt{PLR}(u) = \mathtt{PLR}(C[\![u]\!]) + \mathtt{LM}_x(C[\![u]\!]) \cdot \mathtt{PLR}(u)$.

The proof proceeds by induction on $|C|$.

- If $C = \Box$, then
  $\mathtt{PLR}(C[\![x^\alpha]\!]) + \mathtt{LM}_x(C[\![x^\alpha]\!]) \cdot \mathtt{PLR}(u) = \mathtt{PLR}(C[\![u]\!]) + \mathtt{LM}_x(C[\![u]\!]) \cdot \mathtt{PLR}(u) = \mathtt{PLR}(u)$.
- If $C = \lambda y.C_1$, then $C[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$ implies $y \neq x$. A straightforward inductive argument suffices to conclude.

- If $C = C_1 s$ and $C_1[\![x^\alpha]\!], s \in \mathcal{T}_{\mathcal{WL}}$, then we have

$$\begin{aligned}&\mathtt{PLR}(C[\![x^\alpha]\!]) + \mathtt{LM}_x(C[\![x^\alpha]\!]) \cdot \mathtt{PLR}(u)\\ &\quad= \mathtt{PLR}(C_1[\![x^\alpha]\!]) + \mathtt{PLR}(s) + (\mathtt{LM}_x(C_1[\![x^\alpha]\!]) + \mathtt{LM}_x(s)) \cdot \mathtt{PLR}(u)\\ &\quad= \mathtt{PLR}(C_1[\![x^\alpha]\!]) + \mathtt{LM}_x(C_1[\![x^\alpha]\!]) \cdot \mathtt{PLR}(u) + \mathtt{PLR}(s) + \mathtt{LM}_x(s) \cdot \mathtt{PLR}(u)\end{aligned}$$

  Analogously we obtain

$$\begin{aligned}&\mathtt{PLR}(C[\![u]\!]) + \mathtt{LM}_x(C[\![u]\!]) \cdot \mathtt{PLR}(u)\\ &\quad= \mathtt{PLR}(C_1[\![u]\!]) + \mathtt{LM}_x(C_1[\![u]\!]) \cdot \mathtt{PLR}(u) + \mathtt{PLR}(s) + \mathtt{LM}_x(s) \cdot \mathtt{PLR}(u)\end{aligned}$$

  Hence, IH suffices to conclude.

- Assume $C = C_1 s$, $C_1[\![x^\alpha]\!] = (\lambda z^\beta.s_1)\mathtt{L}$, $(\lambda z.s_1)\mathtt{L} \in \mathcal{T}_{\mathcal{WL}}$, $s \in \mathcal{T}_{\mathcal{WL}}$. Let us define C_1' as in the analogous case in the proof of Lem. 4.2.9, so that $C_1'[\![x^\alpha]\!] = (\lambda z.s_1)\mathtt{L}$. Observe that $\mathtt{PLR}(C[\![x^\alpha]\!]) = 1 + \mathtt{PLR}(C_1'[\![x^\alpha]\!]) + \mathtt{PLR}(s)$, $\mathtt{LM}_x(C[\![x^\alpha]\!]) = \mathtt{LM}_x(C_1'[\![x^\alpha]\!]) + \mathtt{LM}_x(s)$, and analogously for $C[\![u]\!]$. Moreover, IH applies to C_1'. Hence, an argument similar to that used in the previous case applies.

- If $C = sC_2$, so that $C_2[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$, then let us define s' as in the analogous case in the proof of Lem. 4.2.9, so that $\mathtt{PLR}(C[\![x^\alpha]\!]) = 1 + \mathtt{PLR}(s') + \mathtt{PLR}(C_2[\![x^\alpha]\!])$ and $\mathtt{LM}_x(C[\![x^\alpha]\!]) = \mathtt{LM}_x(s') + \mathtt{LM}_x(C_2[\![x^\alpha]\!])$. An argument similar to that of the third case in this proof, considering s', applies.

- If $C = C_1[y/s]$, so that $C_1[\![x^\alpha]\!], s \in \mathcal{T}_{\mathcal{WL}}$, then the occurrences of y in $C[t]$ are bound for any t, implying $x \neq y$ and $y \notin \mathtt{fv}(u)$. Let us call $\mathsf{s_p} = \mathtt{PLR}(s)$, $\mathsf{s_x} = \mathtt{LM}_x(s)$ and $\mathsf{u_p} = \mathtt{PLR}(u)$. Then

$$\begin{aligned}&\mathtt{PLR}(C[\![x^\alpha]\!]) = \mathtt{PLR}(C_1[\![x^\alpha]\!]) + \mathsf{s_p} + \mathtt{LM}_y(C_1[\![x^\alpha]\!]) \cdot \mathsf{s_p} + \mathtt{LM}_y(C_1[\![x^\alpha]\!])\\ &\mathtt{LM}_x(C[\![x^\alpha]\!]) \cdot \mathsf{u_p} = (\mathtt{LM}_x(C_1[\![x^\alpha]\!]) + \mathsf{s_x} + \mathtt{LM}_y(C_1[\![x^\alpha]\!]) \cdot \mathsf{s_x}) \cdot \mathsf{u_p}\end{aligned}$$

  Therefore

$$\begin{aligned}&\mathtt{PLR}(C[\![x^\alpha]\!]) + \mathtt{LM}_x(C[\![x^\alpha]\!]) \cdot \mathsf{u_p}\\ &\quad= \mathtt{PLR}(C_1[\![x^\alpha]\!]) + \mathtt{LM}_x(C_1[\![x^\alpha]\!]) \cdot \mathsf{u_p} + \mathtt{LM}_y(C_1[\![x^\alpha]\!]) \cdot (\mathsf{s_x} \cdot \mathsf{u_p} + \mathsf{s_p} + 1) + \mathsf{s_p} + \mathsf{s_x} \cdot \mathsf{u_p}\end{aligned}$$

  and analogously

$$\begin{aligned}&\mathtt{PLR}(C[\![u]\!]) + \mathtt{LM}_x(C[\![u]\!]) \cdot \mathsf{u_p}\\ &\quad= \mathtt{PLR}(C_1[\![u]\!]) + \mathtt{LM}_x(C_1[\![u]\!]) \cdot \mathsf{u_p} + \mathtt{LM}_y(C_1[\![u]\!]) \cdot (\mathsf{s_x} \cdot \mathsf{u_p} + \mathsf{s_p} + 1) + \mathsf{s_p} + \mathsf{s_x} \cdot \mathsf{u_p}\end{aligned}$$

  Moreover, Lem. 4.2.9 implies $\mathtt{LM}_y(C_1[\![x^\alpha]\!]) = \mathtt{LM}_y(C_1[\![u]\!])$. Hence IH suffices to conclude.

- If $C = s[y/C_2]$, so that $s, C_2[\![x^\alpha]\!] \in \mathcal{T}_{\mathcal{WL}}$, variable convention implies $x \neq y$. We add $\mathsf{s_y} = \mathtt{LM}_y(s)$ to the acronyms used in the previous case. Then

$$\begin{aligned}&\mathtt{PLR}(C[\![x^\alpha]\!]) = \mathsf{s_p} + \mathtt{PLR}(C_2[\![x^\alpha]\!]) + \mathsf{s_y} \cdot \mathtt{PLR}(C_2[\![x^\alpha]\!]) + \mathsf{s_y}\\ &\mathtt{LM}_x(C[\![x^\alpha]\!]) \cdot \mathsf{u_p} = (\mathsf{s_x} + \mathtt{LM}_x(C_2[\![x^\alpha]\!]) + \mathsf{s_y} \cdot \mathtt{LM}_x(C_2[\![x^\alpha]\!])) \cdot \mathsf{u_p}\end{aligned}$$

  Therefore

$$\begin{aligned}&\mathtt{PLR}(C[\![x^\alpha]\!]) + \mathtt{LM}_x(C[\![x^\alpha]\!]) \cdot \mathsf{u_p}\\ &\quad= \mathtt{PLR}(C_2[\![x^\alpha]\!]) + \mathtt{LM}_x(C_2[\![x^\alpha]\!]) \cdot \mathsf{u_p} +\\ &\qquad \mathsf{s_y} \cdot \mathtt{PLR}(C_2[\![x^\alpha]\!]) + \mathsf{s_y} \cdot \mathtt{LM}_x(C_2[\![x^\alpha]\!]) \cdot \mathsf{u_p} + \mathsf{s_p} + \mathsf{s_y} + \mathsf{s_x} \cdot \mathsf{u_p}\\ &\quad= \mathtt{PLR}(C_2[\![x^\alpha]\!]) + \mathtt{LM}_x(C_2[\![x^\alpha]\!]) \cdot \mathsf{u_p} +\\ &\qquad \mathsf{s_y} \cdot (\mathtt{PLR}(C_2[\![x^\alpha]\!]) + \mathtt{LM}_x(C_2[\![x^\alpha]\!]) \cdot \mathsf{u_p}) + \mathsf{s_p} + \mathsf{s_y} + \mathsf{s_x} \cdot \mathsf{u_p}\end{aligned}$$

  and analogously

$$\begin{aligned}&\mathtt{PLR}(C[\![u]\!]) + \mathtt{LM}_x(C[\![u]\!]) \cdot \mathsf{u_p}\\ &\quad= \mathtt{PLR}(C_2[\![u]\!]) + \mathtt{LM}_x(C_2[\![u]\!]) \cdot \mathsf{u_p} +\\ &\qquad \mathsf{s_y} \cdot (\mathtt{PLR}(C_2[\![u]\!]) + \mathtt{LM}_x(C_2[\![u]\!]) \cdot \mathsf{u_p}) + \mathsf{s_p} + \mathsf{s_y} + \mathsf{s_x} \cdot \mathsf{u_p}\end{aligned}$$

Hence IH suffices to conclude.

- If $C = C_1[y^\beta/s]$, or $C = s[y^\beta/C_2]$, then we obtain $x \neq y$ like in the previous cases. An inductive argument like in the third case allows to conclude. $\square$

C.3 Creation lemma

In this section we give the proof of Lem. 4.2.22, cfr. page 111. We need a previous result.

Lemma C.3.1. *Let C be a context, u a term and x a variable. Let C' and u' be variants of C and u respectively, in which all steps in $C[\![x]\!]$, resp. u, are labeled. Let b be a non-labeled step in $C'[\![u']\!]$. Then $C = D[\Box \mathtt{L}t]$, $u = (\lambda y.u_1)\mathtt{L}_1$, and $b = \langle D, (\lambda y.u_1)\mathtt{L}_1\mathtt{L}t\rangle$, i.e. a $\mathtt{db}$-step.*

Proof. By induction on C. Notice that the hypotheses $C[\![x]\!]$ and $C'[\![u']\!]$ imply that $x \notin \mathtt{bv}(C)$ and $\mathtt{fv}(u) \cap \mathtt{bv}(C) = \varnothing$.

- If $C = \Box$ then $C'[u'] = u'$ and therefore all the steps are labeled.
- If $C = C_1 s$, so that $C' = C_1' s'$, observe that if b is inside $C_1'[u']$ then IH suffices to conclude, and if b is inside s' then it is labeled in $C_1[\![x]\!]$. The only other possible case is $b = \langle \Box, C_1'[\![u']\!]s\rangle$, a $\mathtt{db}$-step, so that $C_1'[\![u']\!] = (\lambda y.u_1')\mathtt{L}'$. In turn, b being not labeled, and therefore not present in $C'[\![x]\!]$, implies that $C_1 = \Box$ or $C_1 = \Box\mathtt{L}_2$, hence $u = (\lambda y.u_1)\mathtt{L}_1$ (and $\mathtt{L} = \mathtt{L}_1\mathtt{L}_2$). Thus we conclude by taking $D = \Box$.
- If $C = sC_2$, so that $C' = s'C_2'$, then b being inside s' implies that it is labeled, and if b is inside $C_2'[\![u']\!]$ then IH suffices to conclude. The only case left, namely $s = (\lambda y.s_1)\mathtt{L}$ and $b = \langle \Box, sC_2'u'\rangle$, implies $\langle \Box, sC_2'x\rangle$ to be a step as well, and thus labeled (so that $s' = (\lambda y^\beta.s_1)\mathtt{L}$).
- If $C' = C_1'[y/s']$, then observe that $y \notin \mathtt{fv}(u)$ and $x \neq y$. Moreover, y not being labeled implies $y \in \mathtt{flv}(C_1')$. If b is inside either $C_1'[\![u]\!]$ or s', then we conclude immediately as in previous cases. Notice that no new y occurrences can appear in C_1', nor the existing occurrences can be erased. Therefore, the only other steps in $C_1'[\![u]\!]$ are the $\mathtt{ls}$-redex corresponding to the occurrences of y in C_1', which are labeled. Thus we conclude.
- If $C' = s'[y/C_2']$, so that $y \in \mathtt{flv}(s')$, then b being inside s' or $C_2'[\![u']\!]$ allows a straightforward argument like in previous cases. Moreover, $s' \in \mathtt{flv}(s')$ prevents $\langle \Box, s'[y/C_2'[\![u']\!]]\rangle$ to be a $\mathtt{gc}$-step. Thus we conclude.
- If $C' = C_1'[y^\beta/s']$, then b being inside $C_1'[\![u']\!]$ or s' allows a straightforward argument like in previous cases. Moreover, observe that hypothesis entails $y \notin \mathtt{fv}(u)$, so that the only step for the substitution on y in $C_1'[\![u']\!]$ is the labeled $\mathtt{gc}$-step. Thus we conclude.
- The cases $C' = s'[y^\beta/C_2']$ and $C' = \lambda y.C_1'$ admit straightforward arguments. $\square$

Proof of Lem. 4.2.22.
We recall the statement.
Let $t \xrightarrow{a} t'$, and $b \in \mathcal{RO}(t')$ such that $\varnothing[\![a]\!]b$. Then one of the following conditions holds (where, for readability, β is used to label the created step)

1. ($\mathtt{db}$ creates a $\mathtt{db}$-step)
   $t = C[((\lambda x^{\alpha}.(\lambda y.s)\mathtt{L}_1)\mathtt{L}_2\, u)\,\mathtt{L}_3 v] \to_{\mathtt{db}} C[(\lambda y^{\beta}.s)\mathtt{L}_1[x/u]\mathtt{L}_2\mathtt{L}_3\, v] = t'$
2. ($\mathtt{db}$ creates a $\mathtt{ls}$-step)
   $t = C[(\lambda x^{\alpha}.D[\![x]\!])\mathtt{L}u] \to_{\mathtt{db}} C[D[\![x^{\beta}]\!][x/u]\mathtt{L}] = t'$
3. ($\mathtt{db}$ creates a $\mathtt{gc}$-step)
   $t = C[(\lambda x^{\alpha}.s)\mathtt{L}\, u] \to_{\mathtt{db}} C[s[x^{\beta}/u]\mathtt{L}] = t'$, where $x \notin \mathtt{fv}(s)$
4. ($\mathtt{ls}$ downward creates a $\mathtt{db}$-step)
   $t = C[D[x^{\alpha}\mathtt{L}_2\, u]\,[x/(\lambda y.s)\mathtt{L}_1]] \to_{\mathtt{ls}} C[D[(\lambda y^{\beta}.s)\mathtt{L}_1\mathtt{L}_2 u]\,[x/(\lambda y.s)\mathtt{L}_1]] = t'$
5. ($\mathtt{ls}$ upward creates a $\mathtt{db}$-step)
   $t = C[x^{\alpha}\mathtt{L}_2\,[x/(\lambda y.s)\mathtt{L}_1]\mathtt{L}_3\, u] \to_{\mathtt{ls}} C[(\lambda y^{\beta}.s)\mathtt{L}_1\mathtt{L}_2\,[x/(\lambda y.s)\mathtt{L}_1]\mathtt{L}_3 u] = t'$
6. ($\mathtt{ls}$ creates a $\mathtt{gc}$-step)
   $t = C[D[\![x^{\alpha}]\!][x/u]] \to_{\mathtt{ls}} C[D[\![u]\!][x^{\beta}/u]] = t'$, where $x \notin \mathtt{fv}(D[\![u]\!])$
7. ($\mathtt{gc}$ creates a $\mathtt{gc}$-step)
   $t = C[D[s[y^{\alpha}/E[\![x]\!]]][x/u]] \to_{\mathtt{gc}} C[D[s][x^{\beta}/u]] = t'$, where $y \notin \mathtt{fv}(s)$ and $x \notin \mathtt{fv}(D[s])$.

We proceed by induction on the context D of a. Let us call r the pattern of a, and define $r \xrightarrow{a'} r'$, where a' is the step corresponding to a in r. To improve readability, we mark the anchors of a and b with the labels α and β respectively in the following.

If $D = \Box$, then we perform a case analysis on a.

- If a is a $\mathtt{db}$-step, so that $t = (\lambda x^{\alpha}.s)\mathtt{L}u$ and $t' = s[x/u]\mathtt{L}$, then the only possible cases for b are the following.
  - If b is inside s, $\mathtt{L}$ or u, then it is not created.
  - If $s = C[\![x^{\beta}]\!]$, then case 2 applies.
  - If the pattern of b is $s[x^{\beta}/u]$, so that $x \notin \mathtt{fv}(s)$, then case 3 applies.
- If a is a $\mathtt{ls}$-step, so that $t = C[\![x^{\alpha}]\!]\,[x/u]$ and $t' = C[\![u]\!]\,[x/u]$, then observe that variable convention implies $x \notin \mathtt{fv}(u)$. We list the only possible cases for b.
  - If b is inside $C[\![u]\!]$, then Lem. C.3.1 implies that case 4 applies.
  - If b is inside u, then it is not created.
  - If b is a $\mathtt{ls}$-step on an occurrence of x in $C[\![u]\!]$, then $x \notin \mathtt{fv}(u)$ implies that b is not created.
  - If the pattern of b is $C[\![u]\!]\,[x^{\beta}/u]$, then case 6 applies.
- If a is a $\mathtt{gc}$-step, so that $t = s\,[x/u]$ and $t' = s$, then we conclude immediately.

If $D = D_1 s$, so that $t = D_1[r]s$ and $t' = D_1[r']s$, then the only possible cases for b are the following.

- If b is inside $D_1[r]$, then IH suffices to conclude.
- If b is inside s, then it is not created.

- The only other possible case is $t' = (\lambda x^\beta.s_1)\mathtt{L}s$, so that $D_1[r'] = (\lambda x^\beta.s_1)\mathtt{L}$. In this case, b being created implies that $D_1 = \Box$ or $D_1 = \Box\mathtt{L}_2$, therefore $r' = (\lambda x.s_1)\mathtt{L}_1$, and also that r is not of the shape $(\lambda x.s_1')\mathtt{L}_1'$.

  If a is a `db`-step, then case 1 applies.

  If a is a `ls`-step, then case 5 applies.

  Finally, a being a `gc`-step would contradict $r' = (\lambda x.s_1)\mathtt{L}_1$ or $r \neq (\lambda x.s_1')\mathtt{L}_1'$.

If $D = sD_2$, then b being internal to s or $D_2[r']$ implies that a b is not created, and that IH suffices, respectively. The only other possible case is $s = (\lambda x.s)\mathtt{L}$, so that the pattern of b is $(\lambda x^\beta.s)\mathtt{L}D_2[r']$; in this case b is not created.

If $D = D_1[x/s]$, then b being internal to $D_1[r']$ or to s admits straightforward arguments, like in the previous case. Moreover, observe that an occurrence of x in $D_1[r']$ cannot be created, therefore the only other possible case is the pattern of b being $D_1[r'][x^\beta/s]$, implying that x occurs in $D_1[r]$ but not in $D_1[r']$. Observing that neither a `db`-step nor a `ls`-step can erase an occurrence of x, we conclude that a is a `gc`-step, and therefore that case 7 applies.

Finally, the cases $D = s[x/D_2]$ and $D = \lambda x.D_1$ admit straightforward arguments. $\Box$

C.4 The box order preserves graphical equivalence

An auxiliary lemma is needed.

Lemma C.4.1. *Let $C[\![x^\beta]\!] \sim_\phi C'[\![x^\beta]\!]$ and $a \in \mathcal{RO}(C[\![x^\beta]\!])$. Then x^β is in the box of a iff it is in the box of $\phi(a)$.*

Proof. By induction in the definition of $\sim$ as the reflexive and transitive closure of $\stackrel{1}{\sim}$. The interesting case is when $C[\![x^\beta]\!] \stackrel{1}{\sim}_{x^\beta} C'[\![x^\beta]\!]$. We prove this case by induction on C. In the following, we label the anchor of a with α, and we mention the conditions for a and $\phi(a)$ to include the occurrence of x^β in its box.

- The case $C = \Box$ would contradict $C[\![x^\beta]\!] \stackrel{1}{\sim}_\phi C'[\![x^\beta]\!]$.
- Assume $C = C_1\,s$.

  If $C_1[\![x^\beta]\!] \stackrel{1}{\sim} C_1'[\![x^\beta]\!]$ or $s \stackrel{1}{\sim} s'$, then a is inside $C_1[\![x^\beta]\!]$, and similarly, $\phi(a)$ is inside $C_1'[\![x^\beta]\!]$. Particularly, if $C_1[\![x^\beta]\!] = (\lambda y^\alpha.s_1)\mathtt{L}$, implying $C_1'[\![x^\beta]\!] = (\lambda y^\alpha.s_1)\mathtt{L}'$, then x^β is neither in the box of a nor in that of $\phi(a)$. Hence, if $C_1[\![x^\beta]\!] \stackrel{1}{\sim} C_1'[\![x^\beta]\!]$ we conclude by IH, and if $s \stackrel{1}{\sim} s'$ we conclude immediately.

  If $C_1[\![x^\beta]\!] = C_2[\![x^\beta]\!][y/s_1]$ and $C'[\![x^\beta]\!] = (C_2[\![x^\beta]\!]\,s)[y/s_1]$, then a is inside $C_2[\![x^\beta]\!]$, and similarly for $\phi(a)$. Therefore, we conclude immediately.

  If $C_1[\![x^\beta]\!] = s_1[y/C_2]$, $C'[\![x^\beta]\!] = (s_1\,s)[y/C_2[\![x^\beta]\!]]$, and $y \notin \mathtt{fv}(s)$, then a is a `ls`-step for an occurrence of y inside s_1, it is the `gc`-step on y if $y \notin \mathtt{fv}(s_1)$, or it is inside $C_2[\![x^\beta]\!]$. To conclude, it suffices to notice that exactly the same conditions hold for $\phi(a)$.

- Assume $C = s\,C_2$.

  If $s \stackrel{1}{\sim} s'$ or $C_2[\![x^\beta]\!] \stackrel{1}{\sim} C_2'[\![x^\beta]\!]$, then $s = (\lambda y^\alpha.s_1)\mathtt{L}$ or a is inside $C_2[\![x^\beta]\!]$. Observe that $t = (\lambda y^\alpha.s_1)\mathtt{L}$ implies $t' = (\lambda y^\alpha.s_1')\mathtt{L}'$ and vice versa. Therefore exactly the same cases correspond to $\phi(a)$. Hence we conclude by IH if $C_2[\![x^\beta]\!] \overset{1}{\sim} C'_2[\![x^\beta]\!]$ and a is inside $C_2[\![x^\beta]\!]$, and immediately in the remaining cases.

If $s = s_1[y/s_2]$, $C'[\![x^\beta]\!] = (s_1\, C_2[\![x^\beta]\!])[y/s_2]$ and $y \notin \mathtt{fv}(C_2[\![x^\beta]\!])$, then $s_1 = (\lambda y^\alpha.s'_1)\mathtt{L}$ or a is inside $C_2[\![x^\beta]\!]$, and exactly the same cases hold for $\phi(a)$, thus we conclude.

- Assume $C = C_1[y/s]$. Lemma hypothesis implies $x \neq y$.

  If $C_1[\![x^\beta]\!] \overset{1}{\sim} C'_1[\![x^\beta]\!]$ or $s \overset{1}{\sim} s'$, then a is inside $C_1[\![x^\beta]\!]$, and $\phi(a)$ is inside $C'_1[\![x^\beta]\!]$. Then we conclude by IH if $C_1[\![x^\beta]\!] \overset{1}{\sim} C'_1[\![x^\beta]\!]$, and immediately if $s \overset{1}{\sim} s'$.

  If $C_1 = C_2[z/s_1]$ and $C'[\![x^\beta]\!] = C_2[\![x^\beta]\!][y/s][z/s_1]$, then both a and $\phi(a)$ are inside $C_2[\![x^\beta]\!]$, thus we conclude.

  If $C_1 = s_1[z/C_2]$, $C'[\![x^\beta]\!] = s_1[y/s][z/C_2[\![x^\beta]\!]]$, $y \notin \mathtt{fv}(C_2[\![x^\beta]\!])$ and $z \notin \mathtt{fv}(s)$, then a is a $\mathtt{ls}$-step on a occurrence of z in s_1, a $\mathtt{gc}$-step on z if $z \notin \mathtt{fv}(s_1)$, or a step in $C_2[\![x^\beta]\!]$. We conclude by observing that exactly the same cases hold for $\phi(a)$.

  If $C_1 = \lambda z.C_2$ and $C'[\![x^\beta]\!] = \lambda z.C_2[\![x^\beta]\!][y/s]$, then both a and $\phi(a)$ are inside $C_2[\![x^\beta]\!]$, thus we conclude immediately.

  If $C_1 = s_1 C_2$, $C'[\![x^\beta]\!] = s_1[y/s]C_2[\![x^\beta]\!]$ and $y \notin \mathtt{fv}(C_2[\![x^\beta]\!])$, then the possible cases for a are $s_1 = (\lambda z^\alpha.s_2)\mathtt{L}$ or a inside $C_2[\![x^\beta]\!]$. We conclude by observing that the possible cases for $\phi(a)$ are exactly the same.

- Assume that $C = s[y/C_1]$.

  If $s \overset{1}{\sim} s'$ or $C_1[\![x^\beta]\!] \overset{1}{\sim} C'_1[\![x^\beta]\!]$, then the possible cases for both a and $\phi(a)$ are: a $\mathtt{ls}$-step for an occurrence of y inside s and s', a $\mathtt{gc}$-step on y if $y \notin \mathtt{fv}(s) = \mathtt{fv}(s')$, or a step inside $C_1[\![x^\beta]\!]$ and $C'_1[\![x^\beta]\!]$. Then we conclude by IH if a is inside $C_1[\![x^\beta]\!]$ and $C_1[\![x^\beta]\!] \overset{1}{\sim} C'_1[\![x^\beta]\!]$, and immediately in the remaining cases.

  If $s = s_1[z/s_2]$, $C'[\![x^\beta]\!] = s_1[y/C_1[\![x^\beta]\!]][z/s_2]$, $y \notin \mathtt{fv}(s_2)$ and $z \notin \mathtt{fv}(C_1[\![x^\beta]\!])$, then the possible cases for a are: a $\mathtt{ls}$-step for an occurrence of y inside s_1, a $\mathtt{gc}$-step on y if $y \notin \mathtt{fv}(s_1)$, or a step inside $C_1[\![x^\beta]\!]$. We conclude by observing that the possible cases for $\phi(a)$ are exactly the same.

  If $s = \lambda z.s_1$ and $C'[\![x^\beta]\!] = \lambda z.s_1[y/C_1[\![x^\beta]\!]]$, then an analysis like that of the previous case applies.

  If $s = s_1 s_2$, $C'[\![x^\beta]\!] = s_1[y/C_1[\![x^\beta]\!]]s_2$ and $y \notin \mathtt{fv}(s_2)$, then an analysis like that of the previous cases applies. Notice that $s_1 = (\lambda z^\alpha.s'_1)\mathtt{L}$ implies that x^β is neither in a nor in $\phi(a)$.

- Assume $C = \lambda y.C_1$.

  If $C_1[\![x^\beta]\!] \overset{1}{\sim} C'_1[\![x^\beta]\!]$, then a and $\phi(a)$ are inside $C_1[\![x^\beta]\!]$ and $C'_1[\![x^\beta]\!]$ respectively, therefore IH suffices to conclude.

  If $C_1 = C_2[z/s]$ and $C'[\![x^\beta]\!] = (\lambda y.C_2[\![x^\beta]\!])[z/s]$, then both a and $\phi(a)$ must be inside $C_2[\![x^\beta]\!]$, hence we conclude immediately.

  If $C_1 = s[z/C_2]$ and $C'[\![x^\beta]\!] = (\lambda y.s)[z/C_2[\![x^\beta]\!]]$, then for both a and $\phi(a)$ the possible cases are: a $\mathtt{ls}$-step on an occurrence of z in s, a $\mathtt{gc}$-step on z if $z \notin \mathtt{fv}(s)$, or a step inside $C_2[\![x^\beta]\!]$. Thus we conclude.

□

Proof of Lem. 4.5.2. We recall the statement.
Let t, u be terms s.t. $t \sim_\phi u$, where ϕ is the bijection described in page 114, cfr. Lem. 4.4.9. Then, ϕ commutes with $\prec_{\mathtt{B}}$, i.e. $a \prec_{\mathtt{B}} b$ iff $\phi(a) \prec_{\mathtt{B}} \phi(b)$.

We prove the following, stronger statement: let $t \sim_\phi u$ and $n \in \mathbb{N}$. Then $a \prec^n_{\mathtt{B}} b$ iff $\phi(a) \prec^n_{\mathtt{B}} \phi b$. We proceed by induction on $\langle n, m \rangle$ where m is the transitivity degree when considering $\sim$ as the reflexive-transitive closure of $\overset{1}{\sim}$. The interesting case is when $n = m = 1$, i.e. to prove that $a \prec^1_{\mathtt{B}} b$ iff $\phi(a) \prec^1_{\mathtt{B}} \phi(b)$, if $t \overset{1}{\sim}_\phi u$. We prove this statement by induction on t.

- The case $t = x$ would contradict $t \overset{1}{\sim} u$.
- Assume $t = t_1 t_2$. There are several cases to analyse.
  - Assume $t_1 \overset{1}{\sim} u_1$ and $u = u_1 t_2$.
    If a and b are inside t_1, so that $\phi(a)$ and $\phi(b)$ are in u_1, then IH suffices to conclude.
    If a and b are inside t_2, then we conclude immediately.
    If a is inside t_1, so that $\phi(a)$ is inside u_1, and b is inside t_2, then it is immediate to conclude that $a \not\prec^1_{\mathtt{B}} b$ and $\phi(a) \not\prec^1_{\mathtt{B}} \phi(b)$.
    If a is inside t_2 and b is inside t_1, then a similar argument applies, yielding $a \not\prec^1_{\mathtt{B}} b$ and $\phi(a) \not\prec^1_{\mathtt{B}} \phi(b)$.
    If $t_1 = (\lambda x^\alpha.s)\mathtt{L}$, implying $u_1 = (\lambda x^\alpha.s')\mathtt{L}'$, then $a \prec^1_{\mathtt{B}} b$ iff $\phi(a) \prec^1_{\mathtt{B}} \phi(b)$ iff b is inside t_2.
    If $t_1 = (\lambda x^\beta.s)\mathtt{L}$, implying $u_1 = (\lambda x^\beta.s')\mathtt{L}'$, then $a \not\prec^1_{\mathtt{B}} b$ and $\phi(a) \not\prec^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.

    A note for the rest of the proof: it is easy to observe that if a and b are inside the same subterm, it is possible to conclude either by IH or immediately, and if a and b are inside different subterms, then so are $\phi(a)$ and $\phi(b)$, implying $a \not\prec_{\mathtt{B}} b$ and $\phi(a) \not\prec_{\mathtt{B}} \phi(b)$. Consequently, we will not mention such cases in the following.
  - Assume $t_2 \overset{1}{\sim} u_2$, so that $u = t_1 u_2$.
    If $t_1 = (\lambda x^\alpha.s)\mathtt{L}$, then $a \prec^1_{\mathtt{B}} b$ iff b is inside t_2 iff $\phi(b)$ is inside u_2 iff $\phi(a) \prec^1_{\mathtt{B}} \phi(b)$.
    If $t_1 = (\lambda x^\beta.s)\mathtt{L}$, then $a \not\prec^1_{\mathtt{B}} b$ and $\phi(a) \not\prec^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.
  - Assume $t_1 = s_1[x/s_2]$, so that $t = s_1[x/s_2]t_2$, $u = (s_1 t_2)[x/s_2]$, and $x \notin \mathtt{fv}(t_2)$.
    If $s_1 = (\lambda y^\alpha.s_1')\mathtt{L}$, then $a \prec^1_{\mathtt{B}} b$ iff $\phi(a) \prec^1_{\mathtt{B}} \phi(b)$ iff b is inside t_2. Recall $x \notin \mathtt{fv}(t_2)$.
    If $s_1 = (\lambda y^\beta.s_1')\mathtt{L}$, then $a \not\prec^1_{\mathtt{B}} b$ and $\phi(a) \not\prec^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.
    If a is a `ls`-step for an occurrence of x in s_1, or $x \notin \mathtt{fv}(s_1)$ and a is the `gc`-step on x, then $a \prec^1_{\mathtt{B}} b$ iff $\phi(a) \prec^1_{\mathtt{B}} \phi(b)$ iff b is inside s_2.
    If b is a `ls`-step for an occurrence of x in s_1, so that $\phi(b)$ is the `ls`-step for the same occurrence of x, then $a \prec^1_{\mathtt{B}} b$ iff $\phi(a) \prec^1_{\mathtt{B}} \phi(b)$ iff a is inside s_1 and that occurrence of x is in its box.
    If $x \notin \mathtt{fv}(s_1)$ and b is the `gc`-step on x, then $a \not\prec^1_{\mathtt{B}} b$ and $\phi(a) \not\prec^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.

- Assume that $t = t_1[x/t_2]$. There are several cases to analyse.

  - Assume $t_1 \overset{1}{\sim} u_1$ and $u = u_1[x/t_2]$.

    If a is a $\mathtt{ls}$-step for an occurrence of x in t_1, so that $t_1 = C[\![x^\alpha]\!]$ and $u_1 = C'[\![x^\alpha]\!]$, or $x \notin \mathtt{fv}(t_1)$ and a is the $\mathtt{gc}$-step for x, then $a <^1_{\mathtt{B}} b$ iff $\phi(a) <^1_{\mathtt{B}} \phi(b)$ iff b is inside t_2.

    If b is a $\mathtt{ls}$-step for an occurrence of x in t_1, so that $t_1 = C[\![x^\beta]\!]$ and $u_1 = C'[\![x^\beta]\!]$, then $a <^1_{\mathtt{B}} b$ iff a is inside $C[\![x^\beta]\!]$ and x^β is in the box of a, and similarly, $\phi(a) <^1_{\mathtt{B}} \phi(b)$ iff $\phi(a)$ is inside $C'[\![x^\beta]\!]$ and x^β is in the box of $\phi(a)$. Hence Lem. C.4.1 allows to conclude.

    If $x \notin \mathtt{fv}(t_1) = \mathtt{fv}(u_1)$ and b is the $\mathtt{gc}$-step for x, then $a \not<^1_{\mathtt{B}} b$ and $\phi(a) \not<^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.

  - Assume $t_2 \overset{1}{\sim} u_2$ and $u = t_1[x/u_2]$.

    If a is a $\mathtt{ls}$-step for an occurrence of x in t_1, or $x \notin \mathtt{fv}(t_1)$ and a is the $\mathtt{gc}$-step for x, then $a <^1_{\mathtt{B}} b$ iff b is inside t_2 iff $\phi(b)$ is inside u_2 iff $\phi(a) <^1_{\mathtt{B}} \phi(b)$.

    If b is a $\mathtt{ls}$-step for an occurrence of x in t_1, then $a <^1_{\mathtt{B}} b$ iff $\phi(a) <^1_{\mathtt{B}} \phi(b)$ iff a is inside t_1 and its box contains that occurrence of x.

    If $x \notin \mathtt{fv}(t_1)$ and b is the $\mathtt{gc}$-step for x, then $a \not<^1_{\mathtt{B}} b$ and $\phi(a) \not<^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.

  - Assume $t_1 = s_1[y/s_2]$, so that $t = s_1[y/s_2][x/t_2]$, $u = s_1[x/t_2][y/s_2]$, $x \notin \mathtt{fv}(s_2)$ and $y \notin \mathtt{fv}(t_2)$.

    If a is a $\mathtt{ls}$-step for an occurrence of y in s_1, or $y \notin \mathtt{fv}(s_1)$ and a is the $\mathtt{gc}$-step for y (recall $y \notin \mathtt{fv}(t_2)$), then $a <^1_{\mathtt{B}} b$ iff $\phi(a) <^1_{\mathtt{B}} \phi(b)$ iff b is inside s_2; notice $x \notin \mathtt{fv}(s_2)$.

    The analysis is analogous if a is a $\mathtt{ls}$-step for an occurrence of x in s_1, or $x \notin \mathtt{fv}(s_1)$ and a is the $\mathtt{gc}$-step for x.

    If b is a $\mathtt{ls}$-step for an occurrence of x or y inside s_1, then $a <^1_{\mathtt{B}} b$ iff $\phi(a) <^1_{\mathtt{B}} \phi(b)$ iff a is inside s_1 and its box contains that occurrence of x or y.

    If $x \notin \mathtt{fv}(s_1)$ and b is the $\mathtt{gc}$-step for x, or analogously for y, then $a \not<^1_{\mathtt{B}} b$ and $\phi(a) \not<^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.

  - Assume $t_1 = \lambda y.s_1$, so that $t = (\lambda y.s_1)[x/t_2]$, and $u = \lambda y.s_1[x/t_2]$.

    Then the interesting cases are where a or b is a $\mathtt{ls}$-step for an occurrence of x in s_1, or the $\mathtt{gc}$-step for x if $x \notin \mathtt{fv}(s_1)$. A simpler version of the analysis of the previous case applies.

  - Assume $t_1 = s_1 s_2$, so that $t = (s_1 s_2)[x/t_2]$, $u = s_1[x/t_2]s_2$, and $x \notin \mathtt{fv}(s_2)$.

    If $s_1 = (\lambda y^\alpha.s_1')\mathtt{L}$, then $a <^1_{\mathtt{B}} b$ iff $\phi(a) <^1_{\mathtt{B}} \phi(b)$ iff b is inside s_2. Recall $x \notin \mathtt{fv}(s_2)$.

    If $s_1 = (\lambda y^\beta.s_1')\mathtt{L}$, then $a \not<^1_{\mathtt{B}} b$ and $\phi(a) \not<^1_{\mathtt{B}} \phi(b)$ for any $a \in \mathcal{RO}(t)$.

    The remaining cases for a and b coincide with those specified for the previous case in the proof, and a similar analysis applies.

- Assume that $t = \lambda x.t_1$.

  If $t_1 \overset{1}{\sim} u_1$ and $u = \lambda x.u_1$, then all the steps are inside t_1 and u_1, so that IH suffices to conclude.

If $t_1 = s_1[y/s_2]$ and $u = (\lambda x.s_1)[y/s_2]$, then the interesting cases for a and b are those involving `ls`-steps or `gc`-steps for y, similar to those described in previous case. We conclude by observing that a similar analysis applies.

□

C.5 The box order enjoys **Context-Freeness**

Proof of Lem. 4.5.8, page 123.
We recall the statement: let $t = (\lambda x^\alpha.t_1)[y_1/s_1]\dots[y_n/s_n]t_2$ and $u = t_1[x/t_2][y_1/s_1]\dots[y_n/s_n]$, so that $t \xrightarrow{a} u$ where a is the redex labeled by α, and $b, c \in \mathcal{RO}(t)$, $b', c' \in \mathcal{RO}(u)$ such that $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Then $b \prec^1_{\mathtt{B}} c$ iff $b' \prec^1_{\mathtt{B}} c'$.

We proceed by a case analysis given $b \prec^1_{\mathtt{B}} c$, resp. $b' \prec^1_{\mathtt{B}} c'$. Particularly, observe that the steps in u consisting of either free occurrences of x in t_1 or the eventual `gc`-step on x, are not residuals of steps in t, therefore they contradict the hypotheses. Notice also that $a \in \{b, c\}$ would contradict the existence of b' or c'.

The list of possible cases of $b \prec^1_{\mathtt{B}} c$ for each possible $c \in \mathcal{RO}(t)$ follows. We consider a variant of t in which the anchors of b and c have the only occurrences of the labels β and γ respectively.

- If the anchor of c is inside t_1, i.e. $c \subseteq t_1$ or y_k^γ occurs free in t_1, then the only possibility is $b \subseteq t_1$ and the label γ occurring in the box of b.
- If the anchor of c is inside t_1, i.e. $c \subseteq s_j$, or y_k^γ occurs free in s_j where $j < k$, then there are three possibilities: $[y_j/s_j]$ being in fact $[y_j^\beta/s_j]$, y_j^β occurring free in t_1 or in some s_i where $i < j$, or $b \subseteq s_j$ and the label γ occurring in the box of b.
- If $[y_j/s_j]$ is in fact $[y_j^\gamma/s_j]$ then there is no redex which nests γ.
- If $c \subseteq t_2$, then the only possibility is $b \subseteq t_2$ and the label γ occurring in the box of b.

To conclude, it suffices to observe that in each case the conditions are preserved in u, and moreover, that these are the only cases for $b' \prec^1_{\mathtt{B}} c'$ if we consider only non-created redexes. If the anchor of c is inside s_j, then recall that variable convention implies $y_j \notin \mathtt{fv}(t_2)$. □

Proof of Lem. 4.5.9, page 124.
We recall the statement: let $t = E[\![x^\gamma]\!][x/s]$, c the γ-labeled step in t, $a, b \subseteq E[\![x^\gamma]\!]$, $b[\![a]\!]b'$, and $c[\![a]\!]c'$. If $a \not\prec_{\mathtt{B}} c$, then $b \prec^1_{\mathtt{B}} c$ iff $b' \prec^1_{\mathtt{B}} c'$.

We consider a variant of t in which the anchors of a, b and c have the only occurrences of the labels α, β and γ respectively.

Before proving the lemma, we give two examples which show that the condition $a \not\prec_{\mathtt{B}} c$ is needed in the statement.

1. $y^\alpha y^\beta[y/x^\gamma][x/s] \xrightarrow{\alpha} x^\gamma y^\beta[y/x^\gamma][x/s] = u$. In this case $b \prec^1_{\mathtt{B}} c$ and $b' \not\prec_{\mathtt{B}} c'$, where c' is the residual of c whose anchor is on the left.
2. $z[w^\beta/y^\alpha][y/x^\gamma][x/s] \xrightarrow{\alpha} z[w^\beta/x^\gamma][y/x^\gamma][x/s] = u$. In this case $b \not\prec^1_{\mathtt{B}} c$, and $b' \prec^1_{\mathtt{B}} c'$, where c' is the residual of c whose anchor is on the left. This case is counterexample also if we change $\prec^1_{\mathtt{B}}$ with $\prec_{\mathtt{B}}$ in the conclusion, since $b \prec_{\mathtt{B}} {}^2c$ and $b' \not\prec_{\mathtt{B}} c''$, where c'' is the residual of c whose anchor is on the right.

Now we develop the proof.

Let us consider $t_0 = E[\![x^\gamma]\!] \xrightarrow{a} u_0$. We assume $a \notin \{b, c\}$, otherwise $a \in \{b, c\}$ would contradict the existence of b' and c'. Moreover, $b = c$ implies $a \not<_{\mathtt{B}} b$, and therefore, by Linearity, we have $b' = c'$, thus the statement of the lemma becomes trivial. Hence we also assume $b \neq c$ hereafter. We now proceed by induction on E by observing that the nesting relations in $E[\![x^\gamma]\!][x/s]$ mentioned in the hypothesis of the lemma coincide with those of the smaller terms of the form $D[\![x^\gamma]\!][x/s]$ on which the IH is applied. Observe that $a \not<_{\mathtt{B}} c$ implies that x^γ occurs exactly once in u_0, by Linearity.

- $E = \Box$ contradicts the existence of $a, b \subseteq E[\![x^\gamma]\!]$.
- If $E = \lambda y.D$, then $a, b \subseteq D[\![x^\gamma]\!]$ so we conclude by the IH
- Assume $E = D\, t_2$.

  If $b, a \subseteq D[\![x^\gamma]\!]$, then we conclude by the IH.

  If $b \subseteq D[\![x^\gamma]\!]$ and $a \subseteq t_2$, then we conclude by observing that $u_0 = D[\![x^\gamma]\!]t'_2$, where $t_2 \xrightarrow{a} t'_2$.

  If $b \subseteq t_2$, then $b \not<^1_{\mathtt{B}} c$ and $b' \not<^1_{\mathtt{B}} c'$ because the label γ does not occur in t_2 in neither t_0 nor u_0.

  If $a \bowtie E[\![x^\gamma]\!]$, and $b \subseteq D[\![x^\gamma]\!]$, i.e. either $D = (\lambda y^\alpha.D')\,\mathtt{L}$ or $D = (\lambda y^\alpha.t_{11})\,\mathtt{L}_1\,[z/D']\,\mathtt{L}_2$, then we just conclude by an analysis similar to that described in the proof of Lem. 4.5.8, observing the form of u_0 and noticing that $z \notin \mathtt{fv}(t_2)$.

  If $b \bowtie E[\![x^\gamma]\!]$ is a db-step, then $b \not<^1_{\mathtt{B}} c$ and $b' \not<^1_{\mathtt{B}} c'$.
- Assume $E = t_1 D$.

  If $b, a \subseteq D[\![x^\gamma]\!]$, then we conclude by the IH.

  If $a \subseteq t_1$ and $b \subseteq D[\![x^\gamma]\!]$, then $u_0 = t'_1 D[\![x^\gamma]\!]$, where $t_1 \xrightarrow{\alpha} t'_1$ and we trivially conclude.

  If $b \subseteq t_1$ and $a \not\bowtie E[\![x^\gamma]\!]$, then $b \not<^1_{\mathtt{B}} c$ and $b' \not<^1_{\mathtt{B}} c'$.

  If $a \bowtie E[\![x^\gamma]\!]$ is a db-step, then we get a contradiction with $a \not<_{\mathtt{B}} c$.

  If $b \bowtie E[\![x^\gamma]\!]$ is a db-step, then $b <^1_{\mathtt{B}} c$ and $b' <^1_{\mathtt{B}} c'$ because the free occurrence of x^γ lies inside the box of b in both t_0 and u_0.
- Assume $E = D[y/t_2]$.

  If $b \subseteq D[\![x^\gamma]\!]$, then there are several cases to consider.

  - If $a \subseteq D[\![x^\gamma]\!]$ then we conclude by the IH
  - If y^α occurs free in $D[\![x^\gamma]\!]$, then an analysis of the relation between the occurrences of x^γ and y^α, and the box of b, yields the following four cases for $D[\![x^\gamma]\!]$: $D'[D_1[x^\gamma, y^\alpha]]$, $D'[D_1[\![x^\gamma]\!], y^\alpha]$, $D'[D_1[\![y^\alpha]\!], x^\gamma]$, and $D'[s, x^\gamma, y^\alpha]$. The box of b is the subterm indicated with the context D_1 in the three former cases, and s in the last one. Notice that $b <^1_{\mathtt{B}} c$ in the former two cases, while $b \not<^1_{\mathtt{B}} c$ for the first and fourth cases. We conclude by observing that a similar analysis applies to u_0, and that $b' <^1_{\mathtt{B}} c'$ in the same cases.
  - If either $a \bowtie t_0$ is a gc-step on y, or $a \subseteq t_2$, then $t_0 = D[\![x^\gamma]\!][y^\alpha/t_2] \xrightarrow{a} D[\![x^\gamma]\!] = u_0$ or $t_0 = D[\![x^\gamma]\!][y/t_2] \xrightarrow{a} D[\![x^\gamma]\!][y/t'_2] = u_0$ respectively, so that it is trivial to conclude.

In any other case, i.e. $b \bowtie t$ or $b \subseteq t_2$, we get that $b \not<^1_{\mathtt{B}} c$ and $b' \not<^1_{\mathtt{B}} c'$ because x^γ does not occur free in the box of b. If $b \subseteq t_2$ and a is a `ls`-step on an occurrence of y in $D[\![x^\gamma]\!]$, so that $t_0 = D'[x^\gamma, y^\alpha][y/t_2]$, then $u_0 = D'[x^\gamma, t_2][y/t_2]$, hence neither of the two residuals of b embeds c'.

- Assume $E = t_1[y/D]$.

  If $b, a \subseteq D[\![x^\gamma]\!]$, then we conclude by the IH

  If $b \subseteq D[\![x^\gamma]\!]$ and $a \subseteq t_1$, then $t_0 = t_1[y/D[\![x^\gamma]\!]] \xrightarrow{a} t'_1[y/D[\![x^\gamma]\!]] = u_0$, where $t_1 \xrightarrow{a} t'_1$ thus a trivial arguments allows to conclude.

  If $b \subseteq t_1$, and $a \subseteq t_1$ or $a \subseteq D[\![x^\gamma]\!]$, then $b \not<^1_{\mathtt{B}} c$; notice that a free occurrence of y being inside the box of b would imply $b <^2_{\mathtt{B}} c$ but not $b <^1_{\mathtt{B}} c$. It is then straightforward to verify that $b' \not<^1_{\mathtt{B}} c'$ given the assumptions made on a, even if u_0 includes two residuals of b.

  If $b \bowtie E[\![x^\gamma]\!]$, and $a \subseteq t_1$ or $a \subseteq D[\![x^\gamma]\!]$, then we first observe that a cannot erase neither b nor c by the hypothesis $b[\![a]\!]b'$ and $c[\![a]\!]c'$. Therefore $b <^1_{\mathtt{B}} c$ and $b' <^1_{\mathtt{B}} c'$, given the assumptions made on a.

  The case $a \bowtie E[\![x^\gamma]\!]$ implies that the label γ occurs in the box of a, which contradicts $a \not<_{\mathtt{B}} c$.

□

Proof of Lem. 4.5.10, page 124.
We recall the statement: let $a, b, c \in \mathcal{RO}(t)$. Assume $a \not<_{\mathtt{B}} c$, $b[\![a]\!]b'$, $c[\![a]\!]c'$ and $b' <^n_{\mathtt{B}} d' <^1_{\mathtt{B}} c'$, where d' is a created redex. Then $b' <^k_{\mathtt{B}} c'$ with $k \leqslant n$.

Let us define $t \xrightarrow{a} t'$. In the following, we consider a variant of t in which the anchors of a and c have the only occurrences of the labels α and γ respectively. Moreover, we will sometimes mark the anchor of d' in t' with a δ, for improved readability.

This lemma is needed because it is not true that $a, c \in \mathcal{RO}(t)$, $a \not<_{\mathtt{B}} c$, $t \xrightarrow{a} t'$, $c[\![a]\!]c'$ and $d' <^1_{\mathtt{B}} c'$ implies that d' is not created, i.e., that $d[\![a]\!]d'$ for some d. A counterexample follows: $(y^\alpha x^\gamma)[y/I][x/t] \xrightarrow{a} (Ix^\gamma)[y/I][x/t]$. The λ-calculus exhibits an analogous behavior, e.g. $(\lambda y^\alpha.y((\lambda x^\gamma.t)s))I \xrightarrow{a} I((\lambda x^\gamma.t)s)$. On the other hand, the case $t = y^\alpha[y/x^\gamma][x/t] \xrightarrow{a} x^\gamma[y^\delta/x^\gamma][x/t] = t'$, is not a counterexample because $a <_{\mathtt{B}} c$.

Now we develop the proof. We analyse the possible cases for d' being created, w.r.t. Lem. 4.2.22.

If any of cases 1, 4, 5 or 7 apply, then we have:

$$t = C[((\lambda x^\alpha.(\lambda y.s_1)\mathtt{L}_1)\mathtt{L}_2\, s_2)\,\mathtt{L}_3 u] \xrightarrow{a} C[(\lambda y^\delta.s_1)\mathtt{L}_1[x/s_2]\mathtt{L}_2\mathtt{L}_3\, u] = t',$$
$$t = C[D[x^\alpha\mathtt{L}_2\, u][x/(\lambda y.s)\mathtt{L}_1]] \xrightarrow{a} C[D[(\lambda y^\delta.s)\mathtt{L}_1\mathtt{L}_2 u][x/(\lambda y.s)\mathtt{L}_1]] = t',$$
$$t = C[x^\alpha\mathtt{L}_2[x/(\lambda y.s)\mathtt{L}_1]\mathtt{L}_3\, u] \xrightarrow{a} C[(\lambda y^\delta.s)\mathtt{L}_1\mathtt{L}_2[x/(\lambda y.s)\mathtt{L}_1]\mathtt{L}_3 u] = t', \text{ or}$$
$$t = C[D[s[y^\alpha/E[\![x]\!]]][x/u]] \xrightarrow{a} C[D[s][x^\delta/u]] = t'.$$

In any of these cases, $d' <^1_{\mathtt{B}} c'$ iff the anchor of c' is inside u. We prove the lemma statement by induction on n where $b' <^n_{\mathtt{B}} d'$. If $b' <^1_{\mathtt{B}} d'$, then the box of b' contains all the pattern of d', notice that the latter is a `db`- or a `gc`-step. Therefore, the box of b' contains particularly the subterm u, implying $b' <^1_{\mathtt{B}} c'$, so that the lemma statement holds in this case. For the inductive case: $b' <^{k+1}_{\mathtt{B}} d'$ implies $b' <^k_{\mathtt{B}} e' <^1_{\mathtt{B}} d'$ for some e'. The case just verified implies $e' <^1_{\mathtt{B}} c'$, which suffices to conclude.

Otherwise, one of the cases 2, 3 or 6 apply, i.e.

$t = C[(\lambda x^{\alpha}.D[\![x]\!])\mathtt{L}u] \xrightarrow{a} C[D[\![x^{\delta}]\!][x/u]\mathtt{L}] = t'$,

$t = C[(\lambda x^{\alpha}.s)\mathtt{L}\,u] \xrightarrow{a} C[s[x^{\delta}/u]\mathtt{L}] = t'$, or

$t = C[D[\![x^{\alpha}]\!][x/u]] \xrightarrow{a} C[D[\![u]\!][x^{\delta}/u]] = t'$.

In any of these cases, $d' \prec^{1}_{\mathtt{B}} c'$ iff the anchor of c' is inside u. But then $a \prec^{1}_{\mathtt{B}} c$, contradicting the hypothesis. Thus we conclude. □

Bibliography

[ABKL14] Beniamino Accattoli, Eduardo Bonelli, Delia Kesner, and Carlos Lombardi. A nonstandard standardization theorem. In Suresh Jagannathan and Peter Sewell, editors, *POPL*, pages 659–670. ACM, 2014.

[Acc11] Beniamino Accattoli. *Jumping around the box: graphical and operational studies on λ-calculus and Linear Logic*. PhD thesis, *La Sapienza* University of Rome, february 2011.

[Acc12] Beniamino Accattoli. An abstract factorization theorem for explicit substitutions. In Ashish Tiwari, editor, *RTA*, volume 15 of *LIPIcs*, pages 6–21. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[ACCL91] Martín Abadi, Luca Cardelli, Pierre Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *Journal of Functional Programming*, 4(1):375–416, 1991.

[AK10] Beniamino Accattoli and Delia Kesner. The structural *lambda*-calculus. In Anuj Dawar and Helmut Veith, editors, *CSL*, volume 6247 of *Lecture Notes in Computer Science*, pages 381–395. Springer, 2010.

[AM96] S. Antoy and A. Middeldorp. A sequential reduction strategy. *Theor. Comput. Sci.*, 165(1):75–95, 1996.

[AMR06] Ariel Arbiser, Alexandre Miquel, and Alejandro Ríos. A lambda-calculus with constructors. In Frank Pfenning, editor, *RTA*, volume 4098 of *Lecture Notes in Computer Science*, pages 181–196. Springer, 2006.

[AMR09] Ariel Arbiser, Alexandre Miquel, and Alejandro Ríos. The lambda-calculus with constructors: Syntax, confluence and separation. *J. Funct. Program.*, 19(5):581–631, 2009.

[AN80] André Arnold and Maurice Nivat. The metric space of infinite trees. algebraic and topological properties. *Fundamenta Informaticae*, 3(4):445–476, 1980.

[Bal10a] T. Balabonski. On the implementation of dynamic patterns. In E. Bonelli, editor, *Proceedings of the Fifth International Workshop on Higher-Order Rewriting (HOR)*, number 49, pages 16–30. Electronic Proceedings in Theoretical Computer Science, July 2010. `http://eptcs.org/content.cgi?HOR2010`.

[Bal10b] T. Balabonski. Optimality for dynamic patterns: Extended abstract. In M. Fernández T. Kutsia, W. Schreiner, editor, *Proceedings of the 12th International Conference on Principles and Practice of Declarative Programming (PPDP)*, pages 16–30. ACM, July 2010.

[Bar84] H.P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. Elsevier, Amsterdam, 1984.

[BBLRD96] Zine-El-Abidine Benaissa, Daniel Briaud, Pierre Lescanne, and Jocelyne Rouyer-Degli. $\lambda\nu$, a calculus of explicit substitutions which preserves strong normalisation. *J. Funct. Program.*, 6(5):699–722, 1996.

[BKdV00] Inge Bethke, Jan Willem Klop, and Roel C. de Vrijer. Descendants and origins in term rewriting. *Inf. Comput.*, 159(1-2):59–124, 2000.

[BKdV03] Marc Bezem, Jan Willem Klop, and Roel de Vrijer, editors. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2003.

[BKLR12] Eduardo Bonelli, Delia Kesner, Carlos Lombardi, and Alejandro Ríos. Normalisation for dynamic pattern calculi. In *23rd International Conference on Rewriting Techniques and Applications RTA'12*, volume 15 of *LIPIcs*, pages 117–132. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, Cambridge, UK, 1998.

[BR95] Roel Bloo and Kristoffer Rose. Preservation of strong normalization in named lambda calculi with explicit substitution and garbage collection. In *Computing Science in the Netherlands*, pages 62–72. Netherlands Computer Science Research Foundation, 1995.

[Bru08] H.J. Sander Bruggink. *Equivalence of reductions in higher-order rewriting*. PhD thesis, Utrecht University, 2008.

[CF58] Haskell B. Curry and Robert Feys. *Combinatory Logic*. North-Holland Publishing Company, Amsterdam, 1958.

[CF07] Horatiu Cirstea and Germain Faure. Confluence of pattern-based calculi. In Franz Baader, editor, *RTA*, volume 4533 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2007.

[Chu32] Alonzo Church. A set of postulates for the foundation of logic. *Annals of Mathematics, Second Series*, 33(2):346–366, 1932.

[Chu41] Alonzo Church. *The Calculi of Lambda-Conversion*. Princeton University Press, Princeton, 1941.

[CK98] H. Cirstea and C. Kirchner. ρ-calculus, the rewriting calculus. In *Proceedings of Annual Workshop of the Constraints in Computational Logics Working Group, CCL'98*, Jerusalem (Israel), September 1998.

[CK01] H. Cirstea and C. Kirchner. The rewriting calculus - Part I and Part II. *Logic Journal of the IGPL*, 9(3), 2001.

[CK04] Serenella Cerrito and Delia Kesner. Pattern matching as cut elimination. *Theor. Comput. Sci.*, 323(1-3):71–127, 2004.

[Cou83] Bruno Courcelle. Fundamental properties of infinite trees. *Theor. Comput. Sci.*, 25:95–169, 1983.

[CR36] Alonzo Church and J.B. Rosser. Some properties of conversion. *Transactions of the American Mathematical Society*, 39(3):472–482, 1936.

[Cra09] Karl Crary. A simple proof of call-by-value standardization. Technical Report CMU-CS-09-137, Carnegie-Mellon University, 2009.

[Cza14] Łukasz Czajka. A coinductive confluence proof for infinitary lambda-calculus. In *25rd International Conference on Rewriting Techniques and Applications RTA'14 (to appear)*, Lecture Notes in Computer Science. Springer, 2014.

[dCPdF11] Daniel de Carvalho, Michele Pagani, and Lorenzo Tortora de Falco. A semantic measure of the execution time in linear logic. *Theor. Comput. Sci.*, 412(20):1884–1902, 2011.

[DG01] René David and Bruno Guillaume. A lambda-calculus with explicit weakening and explicit substitution. *Mathematical Structures in Computer Science*, 11(1):169–206, 2001.

[DKP91] Nachum Dershowitz, Stéphane Kaplan, and David A. Plaisted. Rewrite, rewrite, rewrite, rewrite, rewrite, . . *Theor. Comput. Sci.*, 83(1):71–96, 1991.

[EGH+10] Jörg Endrullis, Clemens Grabmayer, Dimitri Hendriks, Jan Willem Klop, and Vincent van Oostrom. Unique normal forms in infinitary weakly orthogonal rewriting. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications, RTA 2010*, volume 6 of *LIPIcs*, pages 85–102. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[EHH+13] Jörg Endrullis, Helle Hvid Hansen, Dimitri Hendriks, Andrew Polonsky, and Alexandra Silva. A coinductive treatment of infinitary rewriting. Presented at WIR 2013, First International Workshop on Infinitary Rewriting, Eindhoven, Netherlands, June 2013, 2013.

[EHK12] Jörg Endrullis, Dimitri Hendriks, and Jan Willem Klop. Highlights in infinitary rewriting and lambda calculus. *Theor. Comput. Sci.*, 464:48–71, 2012.

[For02] Julien Forest. A weak calculus with explicit operators for pattern matching and substitution. In *Proceeding of Rewriting Techniques and Applications (RTA), 13th International Conference*, volume 2378 of *Lecture Notes in Computer Science*, pages 174–191. Springer, 2002.

[For03] Thomas Forster. *Logic, Induction and Sets*, volume 56 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, UK, 2003.

[Gal86] Jean H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*. Harper & Row, 1986.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

[GKK05] John R. W. Glauert, Delia Kesner, and Zurab Khasidashvili. Expression reduction systems and extensions: An overview. In A. Middeldorp, V. van Oostrom, F. van Raamsdonk, and R.C. de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity*, volume 3838 of *Lecture Notes in Computer Science*, pages 496–553. Springer, 2005.

[GLM92] Georges Gonthier, Jean-Jacques Lévy, and Paul-André Melliès. An abstract standardisation theorem. In *LICS*, pages 72–81, 1992.

[Hil96] Barney P. Hilken. Towards a proof theory of rewriting: The simply typed 2lambda-calculus. *Theor. Comput. Sci.*, 170(1-2):407–444, 1996.

[Hin69] R. Hindley. An abstract form of the church-rosser theorem. i. *Journal of Symbolic Logic*, 34(4):545–560, 1969.

[Hin78] R. Hindley. Reductions of residuals are finite. *Transactions of the American Mathematical Society*, 240:345–361, 1978.

[HL89] T. Hardin and J.-J. Lévy. A confluent calculus of substitutions. In *France-Japan Artificial Intelligence and Computer Science Symposium*, 1989.

[HL91] Gérard P. Huet and Jean-Jacques Lévy. Computations in orthogonal rewriting systems, i and ii. In *Computational Logic - Essays in Honor of Alan Robinson*, pages 395–414, 1991.

[Hue80] Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.

[Jay04] C. Barry Jay. The pattern calculus. *ACM Trans. Program. Lang. Syst.*, 26(6):911–937, 2004.

[Jay09] B. Jay. *Pattern Calculus: Computing with Functions and Structures*. Springer Publishing Company, Incorporated, 2009.

[JK06a] B. Jay and D. Kesner. Pure pattern calculus. In P. Sestoft, editor, *ESOP*, volume 3924 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 2006.

[JK06b] C. Barry Jay and Delia Kesner. Patterns as first-class citizens. Available at `http://hal.archives-ouvertes.fr/hal-00229331/fr`, 2006.

[JK09] C. Barry Jay and Delia Kesner. First-class patterns. *J. Funct. Program.*, 19(2):191–225, 2009.

[Kah03] W. Kahl. Basic pattern matchin gcalculi: Syntax, reduction, confluence, and normalisation. Technical Report 16, Software Quality Research Laboratory, McMaster University, 2003.

[Kah04] W. Kahl. Basic pattern matching calculi: A fresh view on matching failure. In Yukiyoshi Kameyama and Peter Stuckey, editors, *Functional and Logic Programming, Proceedings of FLOPS 2004*, volume 2998 of *LNCS*, pages 276–290. Springer-Verlag, April 2004.

[Kah10] Stefan Kahrs. Infinitary rewriting: Foundations revisited. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications, RTA 2010*, volume 6 of *LIPIcs*, pages 161–176. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[Kas00] Ryo Kashima. A proof of the standardization theorem in λ-calculus. Research Reports on Mathematical and Computing Sciences C-145, Tokyo Institute of Technology, 2000.

[KB70] D.E. Knuth and P.B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297, Oxford, 1970. Pergamon Press.

[KdV05] Jan Willem Klop and Roel C. de Vrijer. Infinitary normalization. In *We Will Show Them: Essays in Honour of Dov Gabbay*, volume 2, pages 169–192. College Publications, 2005.

[Ken89] Richard Kennaway. Sequential evaluation strategies for parallel-or and related reduction systems. *Ann. Pure Appl. Logic*, 43(1):31–56, 1989.

[Ken92] Richard Kennaway. On transfinite abstract reduction systems. Technical Report CS-R9205, Centrum voor Wiskunde en Informatica, Netherlands, 1992.

[Kes07] Delia Kesner. The theory of calculi with explicit substitutions revisited. In *CSL*, pages 238–252, 2007.

[Ket12] Jeroen Ketema. Reinterpreting compression in infinitary rewriting. In Ashish Tiwari, editor, *23rd International Conference on Rewriting Techniques and Applications (RTA'12) , RTA 2012, May 28 - June 2, 2012, Nagoya, Japan*, volume 15 of *LIPIcs*, pages 209–224. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[KG97] Zurab Khasidashvili and John R. W. Glauert. The geometry of orthogonal reduction spaces. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 649–659. Springer, 1997.

[KKSdV90] Richard Kennaway, Jan Willem Klop, M. Ronan Sleep, and Fer-Jan de Vries. Transfinite reductions in orthogonal term rewriting systems. Technical Report CS-R9041, Centrum voor Wiskunde en Informatica, Netherlands, 1990.

[KKSdV95] Richard Kennaway, Jan Willem Klop, M. Ronan Sleep, and Fer-Jan de Vries. Transfinite reductions in orthogonal term rewriting systems. *Inf. Comput.*, 119(1):18–38, 1995.

[KKvO04] Jeroen Ketema, Jan Willem Klop, and Vincent van Oostrom. Vicious circles in rewriting systems. CWI Technical Report SEN-E 0427, 2004.

[Klo80] Jan Willem Klop. *Combinatory Reduction Systems.* Phd thesis, Utrecht University, 1980.

[KM89] Jan Willem Klop and Aart Middeldorp. Sequentiality in orthogonal term rewriting systems. Technical Report CS-R8932, Centrum voor Wiskunde en Informatica, Netherlands, 1989.

[KM91] Jan Willem Klop and Aart Middeldorp. Sequentiality in orthogonal term rewriting systems. *Journal of Symbolic Computation*, 12(2):161–196, 1991.

[KÓC08] Delia Kesner and Shane Ó Conchúir. Milner's lambda calculus with partial substitutions, 2008. `http://www.pps.univ-paris-diderot.fr/~kesner/papers/shortpartial.pdf`.

[KR95] Fairouz Kamareddine and Alejandro Ríos. A λ-calculus à la de bruijn with explicit substitutions. In Manuel V. Hermenegildo and S. Doaitse Swierstra, editors, *PLILP*, volume 982 of *Lecture Notes in Computer Science*, pages 45–62. Springer, 1995.

[Kri90] Jean-Louis Krivine. *Lambda-calcul types et modèles.* Masson, Paris, 1990.

[KvOdV08] Jan Willem Klop, Vincent van Oostrom, and Roel C. de Vrijer. Lambda calculus with patterns. *Theor. Comput. Sci.*, 398(1-3):16–31, 2008.

[LRdV14] Carlos Lombardi, Alejandro Ríos, and Roel de Vrijer. Proof terms for infinitary rewriting. In Gilles Dowek, editor, *25rd Intl. Conference on Rewriting Techniques and Applications, joint with 12th Intl. Conference on Typed Lambda Calculi with Applications. RTA-TLCA'14*, volume 8560 of *Lecture Notes in Computer Science*, pages 303–318. Springer, 2014.

[LV02] Carlos Lombardi and Enrique Vetere. *Estudio de relaciones de reducción en el cálculo-λ puro.* Tesis de Licenciatura en Ciencias de la Computación dirigida por Alejandro Ríos, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2002.

[Mel96] Paul-André Melliès. *Description abstraite des Systèmes de Réécriture.* PhD thesis, Université Paris VII, 1996.

[Mel05] Paul-André Melliès. Axiomatic rewriting theory I: A diagrammatic standardization theorem. In A. Middeldorp, V. van Oostrom, F. van Raamsdonk, and R.C. de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity*, volume 3838 of *Lecture Notes in Computer Science*, pages 554–638. Springer, 2005.

[Mil99] Robin Milner. *Communicating and Mobile Systems: The Pi Calculus.* Cambridge University Press, Cambridge, 1999.

[Mil07a] Robin Milner. Local bigraphs and confluence: Two conjectures. *Electronic Notes in Theoretical Computer Science*, 175(3):65–73, 2007.

[Mil07b] Robin Milner. Local bigraphs and confluence: Two conjectures. *ENTCS*, 175(3):65–73, 2007.

[MN98] Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192:3–29, 1998.

[New42] M.H.A. Newman. On theories with a combinatorial definition of "equivalence". *Annals of Mathematics, Second Series*, 43(2):223–243, 1942.

[Nip91] Tobias Nipkow. Higher-order critical pairs. In *Proc. 6th IEEE Symp. Logic in Computer Science*, pages 342–349. IEEE Press, 1991.

[O'D77] M. J. O'Donnell. *Computing in Systems Described by Equations*, volume 58 of *LNCS*. Springer-Verlag, 1977.

[Plo75] Gordon Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.

[Plo77] Gordon D. Plotkin. Lcf considered as a programming language. *Theor. Comput. Sci.*, 5(3):223–255, 1977.

[Ros73] B.K. Rosen. Tree manipulation systems and churc-rosser theorems. *Journal of the ACM*, 20:160–187, 1973.

[Ros92] Kristoffer Høgsbro Rose. Explicit cyclic substitutions. In *Proc. Conditional Term Rewriting Systems, Third International Workshop, CTRS-92, Pont-à-Mousson, France*, pages 36–50, 1992.

[Sch65] David Schroer. *The Church-Rosser theorem.* PhD thesis, Cornell University, Ithaca, USA, 1965.

[Sie65] Wacław Sierpiński. *Cardinal and Ordinal Numbers – second edition revised.* PWN – Polish Scientific Publishers, Warsawa, Polska, 1965.

[SR93] R. C. Sekar and I. V. Ramakrishnan. Programming in equational logic: Beyond strong sequentiality. *Inf. Comput.*, 104(1):78–109, 1993.

[Sup60] Patrick Suppes. *Axiomatic Set Theory.* D. van Nostrand, Princeton, USA, 1960.

[SW01] D. Sangiorgi and D. Walker. *The pi-calculus: a Theory of Mobile Processes.* Cambridge University Press, Cambridge, 2001.

[Tak95] Masako Takahashi. Parallel reductions in lambda-calculus. *Information and Computation*, 118(1):120–127, 1995.

[vO90] V. van Oostrom. Lambda calcuus with patterns. Technical Report IR-228, Vrije Universiteit Amsterdam, 1990.

[vO94] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting.* Phd thesis, Vrije Universiteit Amsterdam, 1994.

[vO99] V. van Oostrom. Normalisation in weakly orthogonal rewriting. In P. Narendran and M. Rusinowitch, editors, *RTA*, volume 1631 of *LNCS*, pages 60–74. Springer-Verlag, 1999.

[vOdV02] Vincent van Oostrom and Roel C. de Vrijer. Four equivalent equivalences of reductions. *Electronic Notes in Theoretical Computer Science*, 70(6):21–61, 2002.

[vOvR14] V. van Oostrom and F. van Raamsdonk. The dynamic pattern calculus as a higher-order pattern rewriting system. In *Proceedings of the Seventh International Workshop on Higher-Order Rewriting (HOR)*. To appear, 2014.

[vR97] F. van Raamsdonk. Outermost-fair rewriting. In P. de Groote, editor, *TLCA*, volume 1210 of *LNCS*, pages 284–299. Springer-Verlag, 1997.

[Zan08] Hans Zantema. Normalization of infinite terms. In Andrei Voronkov, editor, *Rewriting Techniques and Applications, 19th International Conference, RTA 2008*, volume 5117 of *Lecture Notes in Computer Science*, pages 441–455. Springer, 2008.

Index of subjects

The page numbers in *italic* indicate where a concept is defined.

Index of notations – ARS

This index contains the notations defined for the ARS model, the Pure Pattern Calculus, and the linear substitution calculus. Some acronyms related to the λ-calculus are also included.

Index of notations – Proof terms

This index contains the notations defined for the proof term model and for infinitary rewriting.