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

Un cálculo-λ cronometrado

Tesis de Licenciatura en Ciencias de la Computación

Giselle Elizabeth Zeitoune

Director: Dr. Pablo Barenbaum

Buenos Aires, 2023

UN CÁLCULO-λ CRONOMETRADO

Resumen

El cálculo-λ permite estudiar la noción de función computable desde un punto de vista matemático, modelando la abstracción y la aplicación de una función a un argumento. El mecanismo de cómputo por el cual se realiza esta aplicación es denominado β-reducción ($\to_\beta$) y es el principio central del cálculo-λ. En este trabajo extendemos el cálculo-λ a una versión "cronometrada" que llamamos cálculo-$\lambda^\bullet$. Extendemos la sintaxis y la semántica para capturar la noción del costo temporal de computar la aplicación de una función a un argumento. Para esto se incorpora un constructor de términos que representa una demora de una unidad de tiempo, y un operador que modela la acción de esperar hasta que el resultado de un cómputo esté listo. Modificamos el mecanismo de cómputo para que cada paso de β-reducción introduzca una demora. Demostramos que este cálculo preserva propiedades del cálculo-λ, como la confluencia y la normalización fuerte del fragmento tipado. Damos un argumento de terminación débil, bajo ciertas hipótesis de tipabilidad, que exhibe una cota explícita para la longitud de la reducción a forma normal, basado en las características del nuevo cálculo usando la noción de costo definida. Finalmente definimos la noción de término débil y fuertemente *temporizable*, y demostramos que no todos los términos son fuertemente temporizables. Este resultado se obtiene por medio de un sistema de tipos auxiliar en el que los juicios de tipado vienen acompañados de restricciones ecuacionales.

Palabras clave: cálculo-λ, reescritura, sistemas de tipos, semántica operacional, confluencia, terminación.

A TIMED λ-CALCULUS

Abstract

The λ-calculus allows studying the notion of a computable function from a mathematical perspective, modeling the abstraction and application of a function to an argument. The computational mechanism by which this application is performed is called β-reduction ($\to_\beta$) and it is the central principle of the λ-calculus. In this work, we extend the λ-calculus to a "timed" version that we call $\lambda^\bullet$-calculus. We modify the syntax and semantics to capture the notion of the temporal cost of computing the application of a function to an argument.

To achieve this, we introduce a term constructor representing a delay of one unit of time and an operator modeling the action of waiting until the result of a computation is ready. We modify the computation mechanism so that each step of β-reduction introduces a delay. We show that this calculus preserves properties of the λ-calculus, such as confluence and strong normalization of the typed fragment. We provide an argument for weak normalization, under certain typability assumptions, which exhibits an explicit bound for the length of the reduction to normal form based on the characteristics of the new calculus using the defined cost notion.

Finally, we define the notion of weakly and strongly *temporizable* terms and prove that not all terms are strongly temporizable. This result is obtained through an auxiliary type system where typing judgments are accompanied by equational constraints.

Keywords: λ-Calculus, Rewriting, Type Systems, Operational Semantics, Confluence, Termination.

AGRADECIMIENTOS

A todas las personas que me cruce en este camino tan maravilloso que de distintas formas sumaron a hacerlo más divertido, más sencillo y sobretodo más gratificante.

A mi director de tesis, Pablo por todo el tiempo dedicado y la buena predisposición desde el primer momento hasta el final.

A los jurados, Sergio y Rafael por tomarse el tiempo de leer la tesis y estar en la defensa.

A mis compañeros y amigos de la facultad, compañeros de trabajos prácticos que hicieron la carrera entretenida y las horas de estudio pasar volando.

A mis compañeros y amigos de trabajo que supieron darme el lugar para estudiar y rendir siempre que lo necesite.

A mis amigas del secundario y de la vida que siempre estuvieron presentes para festejar pequeños logros.

A mi novio que me hizo compañía mientras preparaba finales y escribía esta tesis.

A mi abuela que celebra cada logro y mi abuelo que siempre me impulso a estudiar.

Y por sobretodo a mi hermana y mis padres que me apoyaron en todo momento, me acompañaron, escucharon y aconsejaron incondicionalmente.

Índice general

1. INTRODUCCIÓN

1.1. El cálculo-λ

El cálculo-λ fue inventado por Alonzo Church en la década de 1930 para estudiar desde un punto de vista matemático la noción de función efectivamente computable, teniendo abstracciones de funciones y la aplicación de una función a un argumento. El cálculo-λ es un área de estudio muy extensa. Uno de los primeros trabajos que aborda su estudio es el de Church [1]. Como material de referencia adicional se puede consultar la monografía de Barendregt [2] o las notas de Selinger [3].

Recordemos brevemente las ideas detrás del cálculo-λ. Si, por ejemplo, tenemos un polinomio $x^2 + 3 \times x + 5$ y nos preguntamos cuál es el valor de la expresión si tenemos $x = 2$, podemos computar esto sustituyendo x por 2 en la expresión: $2^2 + 3 \times 2 + 5$, que luego podemos reducir para obtener la respuesta 15. En cálculo-λ podemos representar esto como:

$$\lambda x.\, (x^2 + 3 \times x + 5)$$

El operador λ nos permite hacer una abstracción sobre x. Intuitivamente, podemos pensar $\lambda x.\, (x^2 + 3 \times x + 5)$ como una expresión esperando un valor para la variable x. El símbolo λ funciona como *ligador* de la variable x, indicando el nombre de la variable que será reemplazada en el cuerpo de la función al aplicarla a un argumento.

Más formalmente, los programas (llamados *términos*) del cálculo-λ se definen inductivamente de tal modo que un término puede ser una *variable* x, una λ*-abstracción* de la forma $\lambda x.\, M$, o una *aplicación* de la forma $M\, N$.

Siguiendo el ejemplo, si aplicáramos dicha función al argumento 2, obtendríamos el siguiente paso de reducción:

$$(\lambda x.\, (x^2 + 3 \times x + 5))\, 2 \;\to\; 2^2 + 3 \times 2 + 5$$

Este mecanismo de cómputo se denomina β*-reducción* y es el principio central del cálculo-λ. Este mecanismo indica de manera general cómo se calcula el resultado de la aplicación de una función a un argumento. En su versión general, la β-reducción tiene la siguiente forma:

$$(\lambda x.\, M)\, N \;\to_\beta\; M\{x := N\}$$

Esto indica que para calcular el resultado de la función $\lambda x.\, M$ aplicada a un argumento arbitrario N se debe proceder a evaluar el cuerpo de la función M, en el que todas las ocurrencias del parámetro formal x fueron reemplazadas por el verdadero argumento N. La notación $M\{x := N\}$ representa el resultado de sustituir las ocurrencias de x en M por N.

Los términos del cálculo-λ con la β-reducción forman un sistema de reescritura *confluente*. La confluencia es una propiedad que, informalmente hablando, expresa que el resultado que se obtiene al computar el valor de una expresión es independiente del orden en el que se realizan los pasos de cómputo. Más formalmente, dado un término M del cálculo-λ, si este reduce en cero o más pasos a dos términos, $M \twoheadrightarrow_\beta N_1$ y $M \twoheadrightarrow_\beta N_2$, entonces existe un término N_3 tal que $N_1 \twoheadrightarrow_\beta N_3$ y $N_2 \twoheadrightarrow_\beta N_3$. Escribimos $\twoheadrightarrow_\beta$ para la relación de β-reducción en cero o más pasos. En otras palabras, la confluencia afirma que si tomamos dos "caminos" de reducción diferentes, siempre hay una manera de juntarlos. Veamos un ejemplo parecido al anterior, donde el argumento ahora es la función identidad aplicada a 2, es decir, $(\lambda x.\, x)\, 2$. Luego, tenemos lo siguiente:

$$(\lambda x.\, (x^2 + 3 \times x + 5))\, ((\lambda y.\, y)\, 2)$$

Podemos ver que valen las siguientes dos reducciones:

$$\begin{tikzcd}
(\lambda x.\, (x^2 + 3 \times x + 5))\, ((\lambda y.\, y)\, 2) \arrow[r] \arrow[d] & (\lambda x.\, (x^2 + 3 \times x + 5))\, 2 \\
((\lambda y.\, y)\, 2)^2 + 3 \times ((\lambda y.\, y)\, 2) + 5 &
\end{tikzcd}$$

Si seguimos reduciendo ambos términos, podemos ver que ambos tienen un reducto en común, en este caso $2^2 + 3 \times 2 + 5$. Este ejemplo sirve como ilustración de que vale la propiedad de confluencia:

$$\begin{tikzcd}
(\lambda x.\, (x^2 + 3 \times x + 5))\, ((\lambda y.\, y)\, 2) \arrow[r] \arrow[d] & (\lambda x.\, (x^2 + 3 \times x + 5))\, 2 \arrow[d] \\
((\lambda y.\, y)\, 2)^2 + 3 \times ((\lambda y.\, y)\, 2) + 5 \arrow[r, two heads] & 2^2 + 3 \times 2 + 5
\end{tikzcd}$$

El cálculo-λ se considera como la base teórica de los lenguajes de programación funcional e influenció en el diseño de varios lenguajes funcionales como Lisp, ML o Haskell. En computación, se usa como herramienta teórica para definir la semántica de lenguajes de programación y estudiar matemáticamente sus propiedades. Hay abundante bibliografía que demuestra que el cálculo-λ resulta útil en un contexto práctico, como núcleo de la implementación de lenguajes de programación funcional[4, 5, 6]. Además, tiene una estrecha conexión con la lógica, lo que permite usarlo como base de sistemas de demostración y como herramienta formal para construir programas correctos. Es importante mencionar que toda función computable puede ser expresada en el cálculo-λ. Esto es lo que hace que sea adecuado para estudiar los fundamentos de los lenguajes de programación.

1.2. Cálculo-λ tipado

El cálculo-λ tiene variantes tipadas, tales como el *cálculo-λ simplemente tipado*. Estas variantes incluyen un sistema de tipado dado por reglas de inferencia que sirven para determinar si un término está bien formado. Un tipo se puede pensar como una especificación de la "naturaleza" o el comportamiento de una entidad computacional, tal como un dato o un programa. Por ejemplo, si una expresión es de tipo `Int`, tenemos la garantía de que evalúa a un número entero, en tanto que si una expresión es de tipo `Int` $\to$ `Int`, tenemos la garantía de que se trata de una función tal que, cuando se le provee un número entero como argumento, produce un número entero como resultado. Cuando decimos que un término está bien formado, nos referimos al requisito de que los programas y los datos se usan correctamente según su tipo. Por ejemplo, un programa no debería sumar un número entero con una función. En el cálculo-λ simplemente tipado, se explicitan los tipos de todas las variables que aparecen en el programa. Técnicamente, se trabaja bajo un contexto de tipado (notado Γ) de la forma $x_1 : A_1, \ldots, x_n : A_n$ que indica que la variable x_i tiene tipo A_i. Siguiendo con el ejemplo anterior, tendríamos que el siguiente término

$$x^2 + 3 \times x + 5$$

tiene tipo `Int` bajo el contexto de tipado x : `Int`, mientras que no tendría tipo bajo el contexto de tipado x : `Int` $\rightarrow$ `Int`. En este ejemplo estamos asumiendo que el tipo `Int` incluye operaciones aritméticas como la suma, la potencia y la multiplicación.

Desde el punto de vista formal, el cálculo-λ simplemente tipado se formula usando tres reglas de tipado:

1. La regla de tipado para las *variables* dice que una variable x tiene tipo A si y sólo si así lo indica el contexto de tipado.

2. La regla de tipado para las *funciones* dice que una λ-abstracción $\lambda x. M$ tiene tipo $A \rightarrow B$ si y sólo si su cuerpo M tiene tipo B bajo la hipótesis de que el parámetro formal x tiene tipo A.

3. La regla de tipado para las *aplicaciones* dice que una aplicación $M\,N$ tiene tipo B si y sólo si la función M es de tipo $A \rightarrow B$ y el argumento N es de tipo A para cierto A.

El cálculo-λ simplemente tipado restringe las expresiones que se evalúan, de tal modo que sólo se permite ejecutar programas bien tipados. Además, es posible asegurar que todo programa bien tipado, con términos bien formados, termina produciendo un valor(es decir, no se cuelga ni produce errores). En particular, el cálculo-λ simplemente tipado cumple propiedades como:

1. **Preservación de tipos.** Si M es un término que tiene tipo A bajo el contexto de tipado Γ y hay un paso de reducción $M \rightarrow M'$, entonces M' tiene el mismo tipo bajo el mismo contexto.

2. **Normalización fuerte.** No existe una secuencia infinita de pasos de reducción $M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow \ldots$.

Ambas propiedades son ampliamente conocidas en el área. Por ejemplo, para una demostración de la propiedad de preservación puede consultarse [7, Proposición 1.2.6] y para una demostración de la propiedad de normalización fuerte puede consultarse [7, Teorema 2.2.1].

1.3. El cálculo-$\lambda^\bullet$

En esta tesis, estudiamos una extensión "cronometrada" del cálculo-λ que llamamos cálculo-$\lambda^\bullet$. Este cálculo extiende la sintaxis y la semántica del cálculo-λ para capturar la noción del "costo" temporal de computar la aplicación de una función a un argumento.

Para cada tipo de datos A, hay un tipo de datos notado $\bullet A$ que corresponde a las expresiones que producen un resultado de tipo A después de una "unidad de tiempo". Por ejemplo, el tipo `Int` corresponde al tipo de los enteros que se encuentran *inmediatamente* disponibles, como `1` o `99`. Por otro lado, el tipo $\bullet$ `Int` corresponde al tipo de los enteros que se encuentran disponibles después de una unidad de tiempo, como (λ`x. x`) `99`, donde $\lambda x. x$ es la función identidad.

Este cálculo incluye una operación de *demora*, notada $\bullet M$, que introduce una demora de una unidad de tiempo antes de producir M como resultado. Además, incluye una operación de *espera*, que permite manipular el valor que produce una expresión que tiene demora. Esto requiere modificar el sistema de tipos para que la aplicación de una función $f : A \to B$ a un argumento $x : A$ sea una expresión de tipo $\bullet B$. Para ello es necesario modificar la semántica operacional de tal modo que cada aplicación introduzca una demora. Siguiendo con el mismo ejemplo, en el cálculo-$\lambda^{\bullet}$, se tiene el siguiente paso de reducción:

$$(\lambda x.\, (x^2 + 3 \times x + 5))\, 2 \ \to \ \bullet (2^2 + 3 \times 2 + 5)$$

1.4. Contribuciones y estructura de la tesis

En este trabajo, definimos precisamente el sistema de tipos y la semántica operacional de la variante "cronometrada" del cálculo-λ, a la que denominamos cálculo-$\lambda^{\bullet}$ y estudiamos sus propiedades metateóricas. En particular, damos una caracterización de las formas normales y probamos las propiedades de preservación de tipos, confluencia y normalización fuerte. Damos dos demostraciones distintas de terminación: la primera de ellas basada en una traducción del cálculo-$\lambda^{\bullet}$ al cálculo-λ y la segunda basada en un argumento aritmético elemental, que aprovecha las características propias de este cálculo "cronometrado". Además, estudiamos la pregunta de cuándo un término es *temporizable*, es decir, dado un término del cálculo-λ, si existe un término correspondiente del cálculo-$\lambda^{\bullet}$.

- En el Capítulo 2 (**Preliminares**) repasamos algunas nociones básicas de reescritura y cálculo-λ que se usan en el resto de la tesis.

- En el Capítulo 3 (**El cálculo-$\lambda^{\bullet}$**) definimos el cálculo-$\lambda^{\bullet}$, explicitando su sintaxis, semántica de reducción y sistema de tipos. Luego caracterizamos las formas normales y demostramos las propiedades de preservación de tipos y confluencia.

- En el Capítulo 4 (**Terminación del cálculo-$\lambda^{\bullet}$**) damos dos demostraciones de terminación del cálculo-$\lambda^{\bullet}$. En primer lugar, damos una demostración por traducción, basándonos en el hecho de que el cálculo-λ simplemente tipado termina. En segundo lugar, damos una cota explícita para la longitud de la reducción a forma normal usando la nueva semántica y sintaxis introducida en el cálculo-$\lambda^{\bullet}$.

- En el Capítulo 5 (**Temporización**) definimos un nuevo sistema de tipos vinculado con el del cálculo-$\lambda^{\bullet}$, al que denominamos cálculo-$\lambda^{?}_{\bullet}$, para estudiar la *temporización* de un término. Obtenemos un resultado de imposibilidad, que afirma que hay términos del cálculo-λ que no es posible temporizar.

- Finalmente, en el Capítulo 6 (**Conclusión**) resumimos las conclusiones del trabajo, mencionamos trabajo relacionado al de esta tesis y describimos posibles líneas de trabajo futuro.

2. NOCIONES PRELIMINARES

En este capítulo introducimos más formalmente conceptos que se utilizan en el resto de la tesis. Primero, en la Sección 2.1 definimos la noción de *sistema de reescritura abstracto* junto con algunas de las propiedades que usamos en la tesis. Luego, en la Sección 2.2 recordamos la sintaxis, semántica y sistema de tipado del cálculo-λ junto con propiedades en las que nos basamos más adelante.

2.1. Sistemas de reescritura

En esta sección presentamos los sistemas de reescritura abstractos junto con algunas propiedades como la confluencia y la normalización.

Definición 2.1.1 (Sistema de reescritura abstracto)**.** Un *sistema de reescritura abstracto* (ARS) es un par $(A, \to)$ donde A es un conjunto, cuyos elementos se llaman *objetos*, y $\to \subseteq A^2$.

- Escribimos $a \to b$ sii $(a, b) \in \to$.
- Escribimos $a \to^n b$ sii a reduce a b en exactamente n pasos. Más precisamente:

$$\begin{aligned} t \to^0 s &\iff t = s \\ t \to^{n+1} s &\iff \exists u. (t \to u) \wedge (u \to^n s) \end{aligned}$$

- Escribimos $a \to^{\leq n} b$ sii a reduce a b en a lo sumo n pasos, es decir, si existe k tal que $0 \leq k \leq n$ y $a \to^k b$.
- Escribimos $\to^*$ o $\twoheadrightarrow$ para la clausura reflexiva–transitiva de $\to$. Es decir, $a \twoheadrightarrow b$ si y sólo si existen un entero $n \geq 0$ y objetos $a_0, \ldots, a_n \in A$ tales que $a = a_0 \to a_1 \to a_2 \to \ldots \to a_n = b$.
- Decimos que un objeto $a \in A$ está en *forma normal* o, equivalentemente, que es *irreducible*, si y sólo si no existe $b \in A$ tal que $a \to b$.

Ejemplo 2.1.2. Un ejemplo de un sistema de reescritura es la relación de orden de los números naturales. Tenemos el ARS $(\mathbb{N}, \to_{\leq})$ y podemos ver que vale

- $1 \to_{\leq} 2$
- $2 \to_{\leq} 3$
- $1 \to_{\leq} 3$

La teoría de la reescritura es un tema extenso, por lo que nos centramos en las propiedades que nos resultan útiles en el resto del trabajo. Para más información sobre el tema, se puede ver el libro de Baader y Nipkow [8] o [9].

Una de las propiedades importantes que puede tener un ARS es la *confluencia*.

Definición 2.1.3 (Confluencia)**.** Un sistema de reescritura abstracto $(A, \to)$ es *confluente* si y sólo si para todo $a, b, c \in A$ tales que $a \twoheadrightarrow b$ y $a \twoheadrightarrow c$ existe $d \in A$ tal que $b \twoheadrightarrow d$ y $c \twoheadrightarrow d$. Gráficamente:

$$\begin{tikzcd} a \arrow[r, two heads] \arrow[d, two heads] & b & \exists d & b \arrow[d, two heads, dotted] \\ c & & c \arrow[r, two heads, dotted] & d \end{tikzcd}$$

Ejemplo 2.1.4. Podemos ver que el ARS $(\mathbb{N}, \to_{\leq})$ es confluente ya que siempre tenemos un número más grande al que ambos números pueden reducir. Por ejemplo, partiendo del 1, podemos reducir a 2 y a 3 y ambos pueden reducir a 4.

$$\begin{tikzcd} 1 \arrow[r, two heads] \arrow[d, two heads] & 2 \arrow[d, two heads, dotted] \\ 3 \arrow[r, two heads, dotted] & 4 \end{tikzcd}$$

El siguiente teorema, que se basa en un argumento de Tait y Martin-Löf, establece condiciones suficientes para la confluencia de un ARS, a partir de tres propiedades específicas. En la tesis, usamos este teorema para probar la confluencia del cálculo-$\lambda^{\bullet}$ (Teorema 3.4.8).

Teorema 2.1.5 (Confluencia vía el argumento de Tait–Martin-Löf)**.** Sea $(A, \to)$ un ARS. Sea $\Rightarrow \subseteq A^2$ que verifica las siguientes propiedades:

1. $\to \subseteq \Rightarrow$
2. $\Rightarrow \subseteq \twoheadrightarrow$
3. Propiedad del diamante: para todo $a, b, c \in A$ tales que $a \Rightarrow b$ y $a \Rightarrow c$ existe $d \in A$ tal que $b \Rightarrow d$ y $c \Rightarrow d$.

Entonces $(A, \to)$ es confluente.

Demostración. Ver [9, Prop. 1.1.11]. □

Otras propiedades importantes de un ARS son las propiedades de *normalización* débil y fuerte.

Definición 2.1.6 (Normalización débil)**.** Un sistema de reescritura abstracto $(A, \to)$ es *débilmente normalizante* (WN) si y sólo si todo objeto reduce a forma normal, es decir, para todo $a \in A$ existe $b \in A$ tal que $a \twoheadrightarrow b$, donde b está en forma normal.

Definición 2.1.7 (Normalización fuerte)**.** Un sistema de reescritura abstracto $(A, \to)$ es *fuertemente normalizante* (SN) si y sólo si no hay secuencias de reducción infinitas, es decir, para todo objeto $a \in A$ no existe una sucesión $a_1, a_2, a_3, \ldots$ tal que $a \to a_1 \to a_2 \to a_3 \to \ldots$.

Observación 2.1.8. Todo ARS fuertemente normalizante también es débilmente normalizante.

Ejemplo 2.1.9 (Sistema WN pero no SN)**.** Puede pasar que un sistema sea débilmente normalizante pero no sea fuertemente normalizante. Por ejemplo, consideremos el ARS $(A, \to)$ donde $A = \{a, b\}$ y la relación de reescritura está dada por:

$$\begin{tikzcd} a \arrow[loop above] \arrow[r] & b \end{tikzcd}$$

Es débilmente normalizante porque $a \twoheadrightarrow b$ y $b \twoheadrightarrow b$, donde en ambos casos b está en forma normal. No es fuertemente normalizante porque $a \to a \to a \to \ldots$. Además, este ARS también es confluente pues para todo objeto $x \in A$ se tiene que $x \twoheadrightarrow b$, de modo que sin importar cuáles sean $x, y, z \in A$ se tendrá:

$$\begin{tikzcd} x \arrow[r, twoheadrightarrow] \arrow[d, twoheadrightarrow] & y \arrow[d, dotted, twoheadrightarrow] \\ z \arrow[r, dotted, twoheadrightarrow] & b \end{tikzcd}$$

2.2. El cálculo-λ

En esta sección recordamos la definición formal del cálculo-λ y algunas de sus propiedades más relevantes. Recordemos primero la sintaxis y las reglas de reducción del cálculo-λ sin tipos.

Definición 2.2.1 (Sintaxis del cálculo-λ)**.** Supongamos dado un conjunto infinito de variables $\{x, y, \ldots\}$. Recordemos el conjunto de *términos* del cálculo-λ que se denota Λ y está dado por la siguiente gramática:

$$\begin{array}{rcll} M, N, \ldots & ::= & x & \text{variables} \\ & \mid & \lambda x.\, M & \text{abstracciones} \\ & \mid & M\, N & \text{aplicaciones} \end{array}$$

Definición 2.2.2 (Reglas de reducción)**.** Recordemos que la relación binaria $\to_\beta$ sobre el conjunto de términos del cálculo-λ es la clausura por compatibilidad con contextos arbitrarios de la siguiente regla de reescritura:

$$(\lambda x.\, M)\, N \quad \to_\beta \quad M\{x := N\}$$

Ejemplo 2.2.3. Veamos un ejemplo de reducción:

$$(\lambda x.\, M)\, ((\lambda y.\, y)\, N) \to_\beta (\lambda x.\, M)\, N \to_\beta M\{x := N\}$$

Como ya mencionamos, el cálculo-λ es confluente:

Teorema 2.2.4. El cálculo-λ es confluente.

Demostración. Ver [2, Teorema 11.1.10]. □

En particular, la confluencia implica la unicidad de formas normales. Es decir, si $M \twoheadrightarrow_\beta N_1$ y $M \twoheadrightarrow_\beta N_2$, donde N_1 y N_2 están en forma normal, entonces $N_1 = N_2$.

Observación 2.2.5. El cálculo-λ no es débilmente normalizante (y por lo tanto tampoco es fuertemente normalizante). Por ejemplo, sea $\Omega = (\lambda x.\, x\, x)\, (\lambda x.\, x\, x)$ y notemos que $\Omega \to \Omega \to \Omega \to \cdots$. Además, no hay otra reducción posible, por lo que no existe un término M en forma normal tal que $\Omega \twoheadrightarrow M$.

2.2.1. El cálculo-λ simplemente tipado

A continuación recordamos también la definición del cálculo-λ simplemente tipado, incluyendo la sintaxis de los tipos y contextos de tipado, así como las reglas de tipado.

Definición 2.2.6 (Sintaxis de los tipos)**.** Supongamos dado un conjunto infinito de tipos base $\{\alpha, \beta, \ldots\}$. El conjunto de *tipos* está dado por la siguiente gramática:

$$A, B, \ldots ::= \alpha \mid A \to B$$

Un *contexto de tipado* es un conjunto de *asignaciones de tipo*, cada una de la forma $x : A$. Asumimos que en un contexto de tipado cada variable aparece a lo sumo una vez. Los contextos de tipado se denotan con letras griegas mayúsculas $(\Gamma, \Delta, \ldots)$. Notamos $\mathsf{dom}(\Gamma)$ al *dominio* de Γ, es decir, al conjunto de variables que aparecen en Γ.

Definición 2.2.7 (Sistema de tipos)**.** Recordemos que los *juicios de tipado* son de la forma $\Gamma \vdash M : A$. Los juicios válidos se obtienen inductivamente a través de las siguientes reglas de tipado:

$$\frac{}{\Gamma, x : A \vdash x : A}\,\textsc{TVar} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.\, M : A \to B}\,\textsc{TAbs} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\, N : B}\,\textsc{TApp}$$

Ejemplo 2.2.8. Usando las reglas de tipado y asumiendo que valen $\Gamma, x : A \vdash M : B$ y $\Gamma \vdash N : A$, tenemos:

$$\pi = \left(\frac{\dfrac{}{\Gamma, y : A \vdash y : A}\,\textsc{TVar}}{\Gamma \vdash \lambda y.\, y : A \to A}\,\textsc{TAbs} \right)$$

$$\frac{\dfrac{\begin{matrix}\vdots \\ \Gamma, x : A \vdash M : B\end{matrix}}{\Gamma \vdash \lambda x.\, M : A \to B}\,\textsc{TAbs} \qquad \dfrac{\begin{matrix}\vdots \\ \pi\end{matrix} \qquad \begin{matrix}\vdots \\ \Gamma \vdash N : A\end{matrix}}{\Gamma \vdash (\lambda y.\, y)\, N : A}\,\textsc{TApp}}{\Gamma \vdash (\lambda x.\, M)\, ((\lambda y.\, y)\, N) : B}\,\textsc{TApp}$$

Una propiedad fundamental del cálculo-λ simplemente tipado es la preservación de tipos. Esta propiedad afirma que, si un término es tipable, todos sus reductos son tipables y, más aún, tienen el mismo tipo. Más precisamente:

Teorema 2.2.9 (Preservación de tipos)**.** Si $\Gamma \vdash M : A$ y $M \to N$ entonces $\Gamma \vdash N : A$.

Demostración. Ver [10, Teorema 8.3.3]. □

Definición 2.2.10 (Cálculo-λ simplemente tipado)**.** El conjunto de *términos tipables* del cálculo-λ se define de la siguiente manera:

$$\Lambda^{\to} \stackrel{\text{def}}{=} \{M \in \Lambda \mid \exists \Gamma, A.\ \Gamma \vdash M : A\}$$

El *cálculo-λ simplemente tipado* es el sistema de reescritura definido sobre el conjunto de términos $\Lambda^{\to}$ con la relación de reescritura $\to$ ya definida.

A diferencia del cálculo-λ no tipado, el cálculo-λ simplemente tipado tiene la propiedad de normalización fuerte, es decir, todos los programas tipables terminan. Más precisamente:

Teorema 2.2.11 (Normalización fuerte)**.** El cálculo-λ simplemente tipado es fuertemente normalizante.

Demostración. Ver [11, Teorema 4.4.6]. $\square$

3. EL CÁLCULO-$\lambda^\bullet$

En este capítulo definimos el cálculo-$\lambda^\bullet$, su sintaxis y semántica y estudiamos algunas de sus propiedades básicas. En la Sección 3.1 damos el conjunto de términos del cálculo-$\lambda^\bullet$ no tipado junto con sus reglas de reducción y algunos lemas que usamos mas adelante. Además, presentamos el sistema de tipos con las reglas de tipado. Luego, en la Sección 3.2 caracterizamos las formas normales del cálculo-$\lambda^\bullet$, dando resultados de correctitud y completitud de la caracterización. En la Sección 3.3 mostramos que se preserva el tipo de un término al reducirlo. Por último, en la Sección 3.4 usamos el argumento de Tait–Martin-Löf para ver que el cálculo-$\lambda^\bullet$ es confluente.

3.1. El cálculo-$\lambda^\bullet$ no tipado

En esta sección damos algunas definiciones y lemas que usamos en todo el trabajo.

Definición 3.1.1 (Sintaxis del cálculo-$\lambda^\bullet$)**.** Suponemos dado un conjunto infinito de variables $\{x, y, \ldots\}$. El conjunto de *términos* del cálculo-$\lambda^\bullet$ se denota $\Lambda_\bullet$ y está dado por la siguiente gramática:

$$\begin{array}{rcll} t, s, \ldots & ::= & x & \text{variables} \\ & \mid & \lambda x.\, t & \text{abstracciones} \\ & \mid & t\, s & \text{aplicaciones} \\ & \mid & \bullet\, t & \text{demora} \\ & \mid & t[x/s] & \text{espera} \end{array}$$

La noción de ocurrencia libre y ligada de una variable en un término se define de la manera usual, considerando que las ocurrencias libres de x en t están ligadas en $\lambda x.\, t$ y en $t[x/s]$. Notamos $\mathsf{fv}(t)$ al conjunto de variables libres de t. Los términos se consideran módulo α-equivalencia, es decir, permitiendo renombrar las variables ligadas.

Definición 3.1.2 (Sustitución sin captura de variables)**.** Escribimos $t\{x := s\}$ para denotar el término que se obtiene de reemplazar todas las ocurrencias libres de x en t por s, evitando la captura de variables. Más precisamente, se puede definir $t\{x := s\}$ por inducción en t del siguiente modo:

$$\begin{array}{rcll} y\{x := s\} & \stackrel{\text{def}}{=} & \begin{cases} s & \text{si } y = x \\ y & \text{si no} \end{cases} & \\ (\lambda y.\, t)\{x := s\} & \stackrel{\text{def}}{=} & \lambda y.\, t\{x := s\} & \text{asumiendo } y \notin \{x\} \cup \mathsf{fv}(s) \\ (t_1\, t_2)\{x := s\} & \stackrel{\text{def}}{=} & t_1\{x := s\}\, t_2\{x := s\} & \\ (\bullet\, t)\{x := s\} & \stackrel{\text{def}}{=} & \bullet\, t\{x := s\} & \\ (t_1[y/t_2])\{x := s\} & \stackrel{\text{def}}{=} & t_1\{x := s\}[y/t_2\{x := s\}] & \text{asumiendo } y \notin \{x\} \cup \mathsf{fv}(s) \end{array}$$

Definición 3.1.3 (Reglas de reducción)**.** Definimos relaciones binarias $\to_{\beta\bullet}, \to_w$ sobre el conjunto de términos del cálculo-$\lambda^\bullet$ como la clausura por compatibilidad con contextos arbitrarios de las siguientes reglas de reescritura:

$$\begin{array}{rcl} (\lambda x.\, t)\, s & \to_{\beta\bullet} & \bullet\, t\{x := s\} \\ t[x/\bullet\, s] & \to_w & \bullet\, t\{x := s\} \end{array}$$

Además, se define la relación $\to$ como la unión de $\to_{\beta\bullet}$ y $\to_w$.

Recordemos que un término t está en *forma normal* (o, equivalentemente, que es *irreducible*) si y sólo si no existe s tal que $t \to s$.

Ejemplo 3.1.4. $(\lambda x.\, t)\, y[y/(\lambda z.\, z)\, s] \to_{\beta\bullet} (\lambda x.\, t)\, y[y/\bullet\, s] \to_w (\lambda x.\, t)\,(\bullet\, s) \to_{\beta\bullet} \bullet\, t\{x := \bullet\, s\}$

Definición 3.1.5 (Reducción en muchos pasos)**.** Se define la relación binaria $\twoheadrightarrow$ como la clausura reflexiva–transitiva de $\to$. Más precisamente:

$$\frac{}{t \twoheadrightarrow t} \qquad \frac{t \to t'}{t \twoheadrightarrow t'} \qquad \frac{t \to t' \quad t' \to t''}{t \twoheadrightarrow t''}$$

Ejemplo 3.1.6. Siguiendo con el Ejemplo 3.1.4, $(\lambda x.\, t)\, y[y/(\lambda z.\, z)\, s] \twoheadrightarrow \bullet\, t\{x := \bullet\, s\}$.

Lema 3.1.7 (La reducción no crea variables)**.** Si $t \to t'$ entonces $\mathsf{fv}(t) \supseteq \mathsf{fv}(t')$.

Demostración. Por inducción en t, observando que $\mathsf{fv}(t\{x := s\}) \subseteq (\mathsf{fv}(t) \setminus \{x\}) \cup \mathsf{fv}(s)$. $\square$

Lema 3.1.8 (Compatibilidad con contextos)**.** Si $t \twoheadrightarrow t'$ entonces:

1. $\lambda x.\, t \twoheadrightarrow \lambda x.\, t'$
2. $t\, s \twoheadrightarrow t'\, s$
3. $s\, t \twoheadrightarrow s\, t'$
4. $\bullet\, t \twoheadrightarrow \bullet\, t'$
5. $t[x/s] \twoheadrightarrow t'[x/s]$
6. $s[x/t] \twoheadrightarrow s[x/t']$

Demostración. Todos los casos son inmediatos por inducción en la derivación de $t \twoheadrightarrow t'$. $\square$

Lema 3.1.9 (Sustitución)**.** Si $x \notin \{y\} \cup \mathsf{fv}(u)$ vale $t\{x := s\}\{y := u\} = t\{y := u\}\{x := s\{y := u\}\}$

Demostración. Por inducción en t.

1. $t = z$:

   1.1 $z = x$, entonces $x\{x := s\}\{y := u\} = s\{y := u\}$. Por otro lado, $x\{y := u\}\{x := s\{y := u\}\} = x\{x := s\{y := u\}\}$ pues $x \neq y$. Podemos ver que $s\{y := u\} = x\{x := s\{y := u\}\}$. Luego $x\{x := s\}\{y := u\} = x\{y := u\}\{x := s\{y := u\}\}$.

   1.2 $z = y$, entonces $y\{x := s\}\{y := u\} = y\{y := u\} = u$. Por otro lado, $y\{y := u\}\{x := s\{y := u\}\} = u\{x := s\{y := u\}\} = u$ pues $x \notin \mathsf{fv}(u)$. Luego $y\{x := s\}\{y := u\} = y\{y := u\}\{x := s\{y := u\}\}$.

   1.3 $z \neq x$ y $z \neq y$, entonces $z\{x := s\}\{y := u\} = z$ y $t\{y := u\}\{x := s\{y := u\}\} = z$. Luego $z\{x := s\}\{y := u\} = z\{y := u\}\{x := s\{y := u\}\}$.

2. $t = \lambda z.\, t'$: asumimos por α-equivalencia $z \neq x$ y $z \neq y$. Por HI $t'\{x := s\}\{y := u\} = t'\{y := u\}\{x := s\{y := u\}\}$. Entonces $\lambda z.\,(t'\{x := s\}\{y := u\}) = \lambda z.\,(t'\{y := u\}\{x := s\{y := u\}\})$. Luego, $(\lambda z.\, t')\{x := s\}\{y := u\} = (\lambda z.\, t')\{y := u\}\{x := s\{y := u\}\}$.

3. $t = t_1\,t_2$: Por HI $t_1\{x := s\}\{y := u\} = t_1\{y := u\}\{x := s\{y := u\}\}$ y $t_2\{x := s\}\{y := u\} = t_2\{y := u\}\{x := s\{y := u\}\}$. Entonces, $(t_1\{x := s\}\{y := u\})(t_2\{x := s\}\{y := u\}) = (t_1\{y := u\}\{x := s\{y := u\}\})(t_2\{y := u\}\{x := s\{y := u\}\})$. Luego, $(t_1\,t_2)\{x := s\}\{y := u\} = (t_1\,t_2)\{y := u\}\{x := s\{y := u\}\}$.

4. $t = \bullet\, t'$: Por HI $t'\{x := s\}\{y := u\} = t'\{y := u\}\{x := s\{y := u\}\}$. Luego, $\bullet\, t'\{x := s\}\{y := u\} = \bullet\, t'\{y := u\}\{x := s\{y := u\}\}$.

5. $t = t_1[z/t_2]$: asumimos por α-equivalencia $z \neq x$ y $z \neq y$. Por HI $t_1\{x := s\}\{y := u\} = t_1\{y := u\}\{x := s\{y := u\}\}$ y $t_2\{x := s\}\{y := u\} = t_2\{y := u\}\{x := s\{y := u\}\}$. Entonces, $t_1\{x := s\}\{y := u\}[z/t_2\{x := s\}\{y := u\}] = t_1\{y := u\}\{x := s\{y := u\}\}[z/t_2\{y := u\}\{x := s\{y := u\}\}]$. Luego, $t_1[z/t_2]\{x := s\}\{y := u\} = t_1[z/t_2]\{y := u\}\{x := s\{y := u\}\}$ pues $z \neq x$ y $z \neq y$.

□

3.1.1. El cálculo-$\lambda^\bullet$ simplemente tipado

A continuación extendemos el cálculo-$\lambda^\bullet$ con una noción de tipo.

Definición 3.1.10 (Sintaxis de los tipos)**.** Supongamos dado un conjunto infinito de tipos base $\{\alpha, \beta, \ldots\}$. El conjunto de *tipos* está dado por la siguiente gramática:

$$A, B, \ldots ::= \alpha \mid A \to B \mid \bullet\, A$$

Un *contexto de tipado* es un conjunto de *asignaciones de tipo*, cada una de la forma $x : A$. Asumimos que en un contexto de tipado cada variable aparece a lo sumo una vez. Los contextos de tipado se denotan con letras griegas mayúsculas $(\Gamma, \Delta, \ldots)$. Notamos $\mathsf{dom}(\Gamma)$ al *dominio* de Γ, es decir, al conjunto de variables que aparecen en Γ.

Definición 3.1.11 (Sistema de tipos)**.** Los *juicios de tipado* son de la forma $\Gamma \rhd t : A$. Los juicios válidos se obtienen inductivamente a través de las siguientes reglas de tipado:

$$\frac{}{\Gamma, x : A \rhd x : A}\ \mathrm{T^\bullet V{\scriptstyle AR}} \qquad \frac{\Gamma, x : A \rhd t : B}{\Gamma \rhd \lambda x.\, t : A \to B}\ \mathrm{T^\bullet A{\scriptstyle BS}} \qquad \frac{\Gamma \rhd t : A \to B \quad \Gamma \rhd s : A}{\Gamma \rhd t\, s : \bullet\, B}\ \mathrm{T^\bullet A{\scriptstyle PP}}$$

$$\frac{\Gamma \rhd t : A}{\Gamma \rhd \bullet\, t : \bullet\, A}\ \mathrm{T^\bullet D{\scriptstyle ELAY}} \qquad \frac{\Gamma, x : A \rhd t : B \quad \Gamma \rhd s : \bullet\, A}{\Gamma \rhd t[x/s] : \bullet\, B}\ \mathrm{T^\bullet W{\scriptstyle AIT}}$$

Ejemplo 3.1.12. Asumiendo que valen $\Gamma, x : \bullet\bullet\, A \rhd t : B$ y $\Gamma \rhd s : A$, tenemos:

$$\pi = \left(\frac{\dfrac{\dfrac{}{\Gamma, z : A \rhd z : A}\ \mathrm{T^\bullet V{\scriptstyle AR}}}{\Gamma \rhd \lambda z.\, z : A \to A}\ \mathrm{T^\bullet A{\scriptstyle BS}} \qquad \dfrac{\vdots}{\Gamma \rhd s : A}}{\Gamma \rhd (\lambda z.\, z)\, s : \bullet\, A}\ \mathrm{T^\bullet A{\scriptstyle PP}} \right)$$

$$\frac{\dfrac{\dfrac{\vdots}{\Gamma, x : \bullet\, A \rhd t : B}}{\Gamma \rhd \lambda x.\, t : \bullet\, A \to B}\ \mathrm{T^\bullet A{\scriptstyle BS}} \qquad \dfrac{\dfrac{}{\Gamma, y : A \rhd y : A}\ \mathrm{T^\bullet V{\scriptstyle AR}} \quad \dfrac{\vdots}{\pi}}{\Gamma \rhd y[y/(\lambda z.\, z)\, s] : \bullet\, A}\ \mathrm{T^\bullet W{\scriptstyle AIT}}}{\Gamma \rhd (\lambda x.\, t)\, y[y/(\lambda z.\, z)\, s] : \bullet\, B}\ \mathrm{T^\bullet A{\scriptstyle PP}}$$

3.2. Caracterización de las formas normales

Recordemos que un término t es *irreducible* (o está en *forma normal*) si no existe un término s tal que $t \to s$. Por ejemplo, $x[x/\lambda y.\, y]$ está en forma normal. Observemos que la noción de forma normal está dada por una condición *negativa*; más concretamente, por la ausencia de subexpresiones reducibles. En general es conceptualmente importante y útil contar también con una caracterización de la noción de forma normal dada por una condición *positiva*, por ejemplo, por la pertenencia a un conjunto definido de manera inductiva.

En esta sección damos una caracterización positiva de las formas normales del cálculo-$\lambda^\bullet$, a través de una gramática (Definición 3.2.1). El resultado principal de esta sección es la Proposición 3.2.4, que afirma que los términos generados por la gramática son irreducibles (Lema 3.2.2) y que todos los términos irreducibles están generados por la gramática (Lema 3.2.3).

Definición 3.2.1. Definimos la siguiente gramática para caracterizar las formas normales del cálculo-$\lambda^\bullet$:

$$\begin{array}{rcl} N & ::= & Q \mid \bullet\, N \mid \lambda x.\, N \\ L & ::= & Q \mid \bullet\, N \\ P & ::= & Q \mid \lambda x.\, N \\ Q & ::= & x \mid N[x/P] \mid L\, N \end{array}$$

Para cada $X \in \{N, L, P, Q\}$ notamos NF_X para el conjunto de términos que se pueden derivar del símbolo no terminal X en la gramática de arriba.

Lema 3.2.2 (Correctitud)**.** Si $t \in \mathsf{NF}_N$, entonces t es irreducible.

Demostración. Demostramos, más en general, que valen las siguientes cuatro implicaciones:

a) Si $t \in \mathsf{NF}_N$, entonces t es irreducible.

b) Si $t \in \mathsf{NF}_L$, entonces t es irreducible y t no es una abstracción.

c) Si $t \in \mathsf{NF}_P$, entonces t es irreducible y t no es una demora.

d) Si $t \in \mathsf{NF}_Q$, entonces t es irreducible y t no es una abstracción ni una demora.

La demostración procede por inducción en la derivación de $t \in \mathsf{NF}_X$ de acuerdo con la gramática de la Definición 3.2.1.

1. Si $t \in \mathsf{NF}_N$ se deriva de $t \in \mathsf{NF}_Q$, entonces por HI (d) concluimos.

2. Si $t \in \mathsf{NF}_N$ donde $t = \bullet\, t'$ con $t' \in \mathsf{NF}_N$, entonces por HI (a) t' es irreducible, luego $\bullet\, t'$ es irreducible pues no hay reglas cuyo lado izquierdo sea de la forma "$\bullet \ldots$".

3. Si $t \in \mathsf{NF}_N$ donde $t = \lambda x.\, t'$ con $t' \in \mathsf{NF}_N$, entonces por HI (a) t' es irreducible, luego $\lambda x.\, t'$ es irreducible pues no hay reglas cuyo lado izquierdo sea de la forma "$\lambda x.\, \ldots$".

4. Si $t \in \mathsf{NF}_L$ se deriva de $t \in \mathsf{NF}_Q$, entonces por HI (d) concluimos.

5. Si $t \in \mathsf{NF}_L$ donde $t = \bullet\, t'$ con $t' \in \mathsf{NF}_N$, entonces por HI (a) t' es irreducible, luego $\bullet\, t'$ es irreducible pues no hay reglas cuyo lado izquierdo sea de la forma "$\bullet \ldots$" y t no es de la forma "$\lambda x.\, t_1$".

6. Si $t \in \mathsf{NF}_P$ se deriva de $t \in \mathsf{NF}_Q$, entonces por HI (d) concluimos.

7. Si $t \in \mathsf{NF}_P$ donde $t = \lambda x.\, t'$ con $t' \in \mathsf{NF}_N$, entonces por HI (a) t' es irreducible, luego $\lambda x.\, t'$ es irreducible pues no hay reglas cuyo lado izquierdo sea de la forma "$\lambda x.\, \ldots$" y t no es de la forma "$\bullet\, t_1$".

8. Si $t \in \mathsf{NF}_Q$ y t es una variable. Luego t es irreducible.

9. Si $t \in \mathsf{NF}_Q$ y $t = t_1[x/t_2]$ con $t_1 \in \mathsf{NF}_N$ y $t_2 \in \mathsf{NF}_P$, por HI (a) t_1 es irreducible y, por HI (c) t_2 es irreducible y no es de la forma $\bullet\, t_2'$. Por lo tanto $t_1[x/t_2]$ no es instancia del lado izquierdo de ninguna regla. Luego $t = t_1[x/t_2]$ es irreducible.

10. Si $t \in \mathsf{NF}_Q$ y $t = t_1\, t_2$ con $t_1 \in \mathsf{NF}_L$ y $t_2 \in \mathsf{NF}_N$, por HI (a) t_2 es irreducible y, por HI (b) t_1 es irreducible y no es de la forma $\lambda x.\, t_1'$. Por lo tanto $t = t_1\, t_2$ no es instancia del lado izquierdo de ninguna regla. Luego $t = t_1\, t_2$ es irreducible.

□

Lema 3.2.3 (Completitud)**.** Si t es irreducible, entonces $t \in \mathsf{NF}_N$.

Demostración. Por inducción en la estructura de t:

1. $t = x$: es inmediato que $x \in \mathsf{NF}_N$.

2. $t = \lambda x.\, t'$: como t es irreducible, t' también lo es. Si no lo fuera, sería posible reducir t' y por lo tanto t sería reducible. Por HI $t' \in \mathsf{NF}_N$ entonces $\lambda x.\, t' \in \mathsf{NF}_N$.

3. $t = t_1\, t_2$: como t es irreducible t_1 y t_2 también lo son. Además, t_1 no es una abstracción. Si lo fuera, $t_1 = \lambda x.\, t_1'$ y entonces $t = (\lambda x.\, t_1')\, t_2$ sería reducible por la regla β. Por HI $t_1 \in \mathsf{NF}_N$ y $t_2 \in \mathsf{NF}_N$. Además, $t_1 \in \mathsf{NF}_L$ pues $t_1 \in \mathsf{NF}_N$ y no es una abstracción. Por lo tanto, $t_1\, t_2 \in \mathsf{NF}_N$.

4. $t = \bullet\, t'$: como t es irreducible, t' también lo es, Por HI $t' \in \mathsf{NF}_N$ entonces $\bullet\, t' \in \mathsf{NF}_N$.

5. $t = t_1[x/t_2]$: como t es irreducible t_1 y t_2 también lo son. Además, t_2 no es una demora. Si lo fuera, $t_2 = \bullet\, t_2'$ y entonces $t = t_1[x/\bullet\, t_2']$ sería reducible por la regla γ. Por HI $t_1 \in \mathsf{NF}_N$ y $t_2 \in \mathsf{NF}_N$. Además, $t_2 \in \mathsf{NF}_P$ pues $t_2 \in \mathsf{NF}_N$ y no es una demora. Por lo tanto, $t_1[x/t_2] \in \mathsf{NF}_N$.

□

Proposición 3.2.4 (Caracterización de formas normales)**.** Sea t un término. Entonces t es irreducible si y sólo si $t \in \mathsf{NF}_N$.

Demostración. Es una consecuencia inmediata de Lemas 3.2.2 y 3.2.3. □

Ejemplo 3.2.5. Los siguientes términos son irreducibles (o, lo que es lo mismo decir, están en forma normal):

1. x

2. $(x\, y)[x/\lambda z.\, z]$

3. $((\lambda x.\, x)[z/y])\, y$

Es fácil ver que estos términos pertenecen al conjunto NF_N.

Por otro lado, podemos ver que los siguientes términos no pertenecen al conjunto de términos que se pueden derivar de la gramática y por lo tanto son reducibles:

1. $x[x/\bullet\, y] \to \bullet\, x\{x := y\}$

2. $(\lambda x.\, x)\, y \to \bullet\, x\{x := y\}$

3.3. Preservación de tipos

Recordemos que la propiedad de preservación de tipos (también conocida como *subject reduction*) afirma que si un término tipable reduce, su reducto también es tipable con el mismo tipo. Esta es una propiedad fundamental, que le da sentido al sistema de tipos. Por ejemplo, sirve para asegurar que, si se ejecuta un programa de tipo `Int`, el resultado final del cómputo también es de tipo `Int`.

En esta sección demostramos la propiedad de preservación de tipos del cálculo-$\lambda^\bullet$. El resultado principal de esta sección es la Proposición 3.3.3. Como resultados auxiliares, necesitamos demostrar propiedades de debilitamiento Lema 3.3.1 y un lema de sustitución Lema 3.3.2.

Lema 3.3.1 (Debilitamiento)**.** Si $\Gamma \rhd t : A$ y $x \notin \mathsf{dom}(\Gamma)$ entonces, $\Gamma, x : B \rhd t : A$.

Demostración. Por inducción en la derivación del juicio $\Gamma \rhd t : A$.

1. $\mathrm{T}^\bullet\textsc{Var}$, $t = y$: sabemos $\Gamma', y : A \rhd y : A$, donde $\Gamma = \Gamma', y : A$. Como $x \notin \mathsf{dom}(\Gamma)$, sabemos que $x \neq y$. Luego, $\Gamma', y : A, x : B \rhd y : A$.

2. $\mathrm{T}^\bullet\textsc{Abs}$, $t = \lambda y.\, t'$: sabemos que $\Gamma \rhd \lambda y.\, t' : C \to D$ con $A = C \to D$ se deriva por $\mathrm{T}^\bullet\textsc{Abs}$ de $\Gamma, y : C \rhd t' : D$. Podemos asumir por α-equivalencia que $y \neq x$. Además, $x \notin \mathsf{dom}(\Gamma)$ y por lo tanto $x \notin \mathsf{dom}(\Gamma, y : C)$. Por HI, $\Gamma, y : C, x : B \rhd t' : D$ y por $\mathrm{T}^\bullet\textsc{Abs}$, $\Gamma, x : B \rhd \lambda y.\, t' : C \to D$.

3. $\mathrm{T}^\bullet\textsc{App}$, $t = su$: sabemos que $\Gamma \rhd su : \bullet\, D$ con $A = \bullet\, D$ se deriva por $\mathrm{T}^\bullet\textsc{App}$ de $\Gamma \rhd s : C \to D$ y $\Gamma \rhd u : C$. Sabemos $x \notin \mathsf{dom}(\Gamma)$. Por HI, $\Gamma, x : B \rhd s : C \to D$ y $\Gamma, x : B \rhd u : C$. Por $\mathrm{T}^\bullet\textsc{App}$, $\Gamma, x : B \rhd s\, u : \bullet\, D$.

4. $\mathrm{T}^\bullet\textsc{Delay}$, $t = \bullet\, t'$: sabemos que $\Gamma \rhd \bullet\, t' : \bullet\, C$ con $A = \bullet\, C$ se deriva por $\mathrm{T}^\bullet\textsc{Delay}$ de $\Gamma \rhd t' : C$. Sabemos $x \notin \mathsf{dom}(\Gamma)$. Por HI, $\Gamma, x : B \rhd t' : C$. Por $\mathrm{T}^\bullet\textsc{Delay}$ $\Gamma, x : B \rhd \bullet\, t' : \bullet\, C$.

5. $\mathrm{T}^\bullet\textsc{Wait}$, $t = s[y/u]$: sabemos que $\Gamma \rhd s[y/u] : \bullet\, C$ con $A = \bullet\, C$ se deriva por $\mathrm{T}^\bullet\textsc{Wait}$ de $\Gamma, y : D \rhd s : C$ y $\Gamma \rhd u : \bullet\, D$. Podemos asumir por α-equivalencia que $y \neq x$. Además, $x \notin \mathsf{dom}(\Gamma)$, luego, $x \notin \mathsf{dom}(\Gamma, y : D)$. Por HI, $\Gamma, y : D, x : B \rhd s : C$ y $\Gamma, x : B \rhd u : \bullet\, D$. Por lo tanto, por $\mathrm{T}^\bullet\textsc{Wait}$, $\Gamma, x : B \rhd s[y/u] : \bullet\, C$.

□

Lema 3.3.2 (Sustitución)**.** Si $\Gamma, x : B \rhd t : A$ y $\Gamma \rhd s : B$ entonces, $\Gamma \rhd t\{x := s\} : A$.

Demostración. Por inducción en la derivación del juicio $\Gamma, x : B \rhd t : A$:

1. $\mathrm{T}^\bullet\textsc{Var}$, $t = y$:

1.1 Si $s \neq t$ entonces, $y\{x := s\} = y$.
Sabemos $\Gamma, x : B \rhd y : A$ y se deriva por $\mathrm{T}^\bullet\textsc{Var}$. Luego, sabemos que $\Gamma = \Gamma', y : A$. Por lo tanto $\Gamma \rhd y : A$.

1.2 Si $s = t$ entonces $y\{x := s\} = s$.
Sabemos $\Gamma, x : B \rhd x : A$ y $\Gamma \rhd s : B$. Como $\Gamma, x : B \rhd x : A$ se deriva de $\mathrm{T}^\bullet\textsc{Var}$, $A = B$. Luego, $\Gamma \rhd s : A$.

2. $\mathrm{T}^\bullet\textsc{Abs}$, $t = \lambda y.\, t'$: asumimos por α-equivalencia que $y \neq x$ y que $y \notin \mathsf{fv}(s)$.

   Sabemos $\Gamma, x : B \rhd \lambda y.\, t' : A$ y se dervia de $\mathrm{T}^\bullet\textsc{Abs}$, entonces, $A = C \to D$ y $\Gamma, x : B, y : C \rhd t' : D$. Usando Lema 3.3.1 sabemos que $\Gamma, y : C \rhd s : B$, y por lo tanto podemos aplicar la HI para concluir que $\Gamma, y : C \rhd t'\{x := s\} : D$. Por $\mathrm{T}^\bullet\textsc{Abs}$, $\Gamma \rhd \lambda y.\, (t'\{x := s\}) : C \to D$. Por definición de la sustitución, $(\lambda y.\, t')\{x := s\} = \lambda y.\, (t'\{x := s\})$ pues $y \neq x$ y $y \notin \mathsf{fv}(s)$.

3. $\mathrm{T}^\bullet\textsc{App}$, $t = u\, r$: sabemos $\Gamma, x : B \rhd u\, r : A$. Como se deriva de $\mathrm{T}^\bullet\textsc{App}$, $\Gamma, x : B \rhd u : C \to D$, $\Gamma, x : B \rhd r : C$ y $A = \bullet D$. Por HI, $\Gamma \rhd u\{x := s\}r\{x := s\} : \bullet D$ y por definición de sustitución $\Gamma \rhd (u\, r)\{x := s\} : \bullet D$.

4. $\mathrm{T}^\bullet\textsc{Delay}$, $t = \bullet\, t'$:

   sabemos $\Gamma, x : B \rhd \bullet\, t' : A$. Como se deriva de $\mathrm{T}^\bullet\textsc{Delay}$, $\Gamma, x : B \rhd t' : C$ y $A = \bullet\, C$. Por HI, $\Gamma \rhd t'\{x := s\} : C$ y por $\mathrm{T}^\bullet\textsc{Delay}$ $\Gamma \rhd \bullet\, (t'\{x := s\}) : \bullet\, C$. Por definición de sustitución, $\Gamma \rhd (\bullet\, t')\{x := s\} : \bullet\, C$.

5. $\mathrm{T}^\bullet\textsc{Wait}$, $t = u[y/r]$: asumimos por α-equivalencia que $y \neq x$ y que $y \notin \mathsf{fv}(s)$.

   Sabemos que $\Gamma, x : B \rhd u[y/r] : A$ y se deriva de $\mathrm{T}^\bullet\textsc{Wait}$ entonces, $\Gamma, x : B, y : C \rhd u : D$ y $\Gamma, x : B \rhd r : \bullet\, C$ con $A = \bullet\, D$. Por Lema 3.3.1 sabemos que $\Gamma, y : C \rhd s : B$, y por lo tanto podemos aplicar la HI sobre la primera premisa para concluir que vale $\Gamma, y : C \rhd u\{x := s\} : D$. Además, por HI, $\Gamma \rhd r\{x := s\} : \bullet\, C$. Por $\mathrm{T}^\bullet\textsc{Wait}$, $\Gamma, y : C \rhd u\{x := s\}[y/r\{x := s\}] : \bullet\, D$ Por definición de sustitución, como $y \neq x$ y $y \notin \mathsf{fv}(s)$, $\Gamma, y : C \rhd u[y/r\{x := s\}] : \bullet\, D$.

□

Proposición 3.3.3 (Preservación de tipos)**.** Si $\Gamma \rhd t : A$ y $t \to t'$ entonces, $\Gamma \rhd t' : A$.

Demostración. Por inducción en la derivación del juicio $\Gamma \rhd t : A$.

1. $\mathrm{T}^\bullet\textsc{Var}$, $t = x$ y no reduce. Luego este caso es inmediato, porque no se cumple el antecedente.

2. $\mathrm{T}^\bullet\textsc{Abs}$, $t = \lambda x.\, s$ y $A = B \to C$: sabemos $\Gamma, x : B \rhd s : C$. Suponemos que $t \to t'$, es decir, $\lambda x.\, s \to t'$. Por lo tanto, la única posibilidad es que $t' = \lambda x.\, s'$ con $s \to s'$. Luego, $\Gamma, x : B \rhd s' : C$ por HI. Por lo tanto, $\Gamma \rhd t' : B \to C$ por $\mathrm{T}^\bullet\textsc{Abs}$.

3. $\mathrm{T}^\bullet\textsc{App}$, $t = s\, u$ y $A = \bullet\, B$: sabemos $\Gamma \rhd s : C \to B$ y $\Gamma \rhd u : C$. Suponemos que $t \to t'$, es decir, $su \to t'$. Hay tres subcasos, dependiendo de si la reducción es interna a s, si es interna a u o en la raíz.

   3.1 $s\, u \to s'\, u$ con $s \to s'$. Luego, $\Gamma \rhd s' : C \to B$ por HI. Por lo tanto, $t' = s'\, u$, $\Gamma \rhd t' : \bullet\, B$ por $\mathrm{T}^\bullet\textsc{App}$.

3.2 $s\,u \to s\,u'$ con $u \to u'$. Luego, $\Gamma \rhd u' : C$ por HI. Por lo tanto, $t' = s\,u'$, $\Gamma \rhd t' : \bullet B$ por T$^\bullet$App.

3.3 $s = \lambda x.\, s'$ y el paso de reducción es de la forma $t = (\lambda x.\, s')\, u \to_{\beta\bullet} \bullet s'\{x := u\} = t'$. Como vale $\Gamma \rhd \lambda x.\, s' : C \to B$ y solo se puede concluir por T$^\bullet$Abs, vale $\Gamma, x : C \rhd s' : B$. Por Lema 3.3.2, $\Gamma \rhd s'\{x := u\} : B$. Por lo tanto, $\Gamma \rhd \bullet s'\{x := u\} : \bullet B$ por T$^\bullet$Delay.

4. T$^\bullet$Delay, $t = \bullet s$ y $A = \bullet B$: sabemos $\Gamma \rhd s : B$. Suponemos que $t \to t'$, es decir, $\bullet s \to t'$. Por lo tanto, la única posibilidad es que $t' = \bullet s'$ con $s \to s'$. Luego, $\Gamma \rhd s' : B$ por HI. Por lo tanto, $\Gamma \rhd t' : \bullet B$ por T$^\bullet$Delay.

5. T$^\bullet$Wait, $t = s[x/u]$ y $A = \bullet B$: sabemos $\Gamma, x : C \rhd s : B$ y $\Gamma \rhd u : \bullet C$. Suponemos que $t \to t'$, es decir, $s[x/u] \to t'$. Hay tres subcasos, dependiendo de si la reducción es interna a s, si es interna a u o en la raíz.

5.1 $s[x/u] \to s'[x/u]$ con $s \to s'$. Luego, $\Gamma, x : C \rhd s' : B$ por HI. Por lo tanto, $t' = s'[x/u]$ y $\Gamma \rhd t' : \bullet B$ por T$^\bullet$Wait.

5.2 $s[x/u] \to s[x/u']$ con $u \to u'$. Luego, $\Gamma \rhd u' : \bullet C$ por HI. Por lo tanto, $t' = s[x/u] \to s[x/u']$ y $\Gamma \rhd t' : \bullet B$ por T$^\bullet$Wait.

5.3 $u = \bullet u'$ y el paso de reducción es de la forma $t = s[x/u] = s[x/(\bullet u')] \to_w \bullet s\{x := u'\} = t'$. Como vale $\Gamma \rhd \bullet u' : \bullet C$ y solo se puede concluir por T$^\bullet$Delay, vale $\Gamma \rhd u' : C$. Además, vale $\Gamma, x : C \rhd s : B$. Luego, por Lema 3.3.2, vale $\Gamma \rhd s\{x := u'\} : B$ Por lo tanto, $\Gamma \rhd \bullet s\{x := u'\} : \bullet B$ por T$^\bullet$Delay.

□

3.4. Confluencia

Recordemos que la confluencia de un sistema de reescritura afirma que el resultado de computar el valor de una expresión es único, independientemente del orden en el que se reduzcan las subexpresiones. En realidad, la definición formal (Definición 2.1.3) de confluencia es un poco más sutil, porque podría pasar que un término no tenga forma normal, o que simultáneamente existan un camino infinito y otro que llega a una forma normal.

En esta sección, demostramos la confluencia del cálculo-$\lambda^\bullet$ usando el argumento de Tait y Martin-Löf (Teorema 2.1.5). Nuestro primer paso es definir una relación $\Rightarrow$ que corresponde a la *reducción en paralelo* (Definición 3.4.1), es decir, una relación que permite reducir varias subexpresiones reducibles en simultáneo. A continuación, demostramos las tres propiedades que permiten aplicar dicho argumento. El Lema 3.4.4 afirma que un paso de reducción se puede simular con un paso de reducción en paralelo. El Lema 3.4.5 afirma que un paso de reducción en paralelo se puede simular con varios pasos de reducción. El Lema 3.4.7 afirma que la reducción en paralelo tiene la propiedad del diamante. Por último, todos estos elementos se combinan para obtener la propiedad de confluencia (Teorema 3.4.8).

Definición 3.4.1 (Reducción en paralelo)**.**

$$\frac{}{x \Rightarrow x}\Rightarrow\textsc{Var} \qquad \frac{t \Rightarrow t'}{\lambda x.\, t \Rightarrow \lambda x.\, t'}\Rightarrow\textsc{Abs} \qquad \frac{t \Rightarrow t' \quad s \Rightarrow s'}{t\, s \Rightarrow t'\, s'}\Rightarrow\textsc{App1}$$

$$\frac{t \Rightarrow t' \quad s \Rightarrow s'}{(\lambda x.\, t)\, s \Rightarrow \bullet\, t'\{x := s'\}} \Rightarrow\textsc{App2} \qquad \frac{t \Rightarrow t'}{\bullet\, t \Rightarrow \bullet\, t'} \Rightarrow\textsc{Delay}$$

$$\frac{t \Rightarrow t' \quad s \Rightarrow s'}{t[x/s] \Rightarrow t'[x/s']} \Rightarrow\textsc{Wait1} \qquad \frac{t \Rightarrow t' \quad s \Rightarrow s'}{t[x/\bullet\, s] \Rightarrow \bullet\, t'\{x := s'\}} \Rightarrow\textsc{Wait2}$$

Ejemplo 3.4.2. Sea $\mathtt{I}$ la identidad, definida como $\mathtt{I} \stackrel{\text{def}}{=} \lambda x.\, x$, tenemos:

$$\pi = \left(\dfrac{\dfrac{\dfrac{}{x \Rightarrow x} \Rightarrow\textsc{Var}}{\mathtt{I} \Rightarrow \mathtt{I}} \Rightarrow\textsc{Abs}}{\bullet\, \mathtt{I} \Rightarrow \bullet\, \mathtt{I}} \Rightarrow\textsc{Delay} \right)$$

$$\dfrac{\dfrac{\dfrac{}{x \Rightarrow x} \Rightarrow\textsc{Var}}{\mathtt{I} \Rightarrow \mathtt{I}} \Rightarrow\textsc{Abs} \qquad \dfrac{\dfrac{}{y \Rightarrow y} \Rightarrow\textsc{Var} \qquad \begin{matrix}\vdots \\ \pi\end{matrix}}{y[y/\bullet\, \mathtt{I}] \Rightarrow \bullet\, \mathtt{I}} \Rightarrow\textsc{Wait2}}{\mathtt{I}\, y[y/\bullet\, \mathtt{I}] \Rightarrow \bullet\, (x\{x := \bullet\, (y\{y := \mathtt{I}\})\}) = \bullet \bullet\, \mathtt{I}} \Rightarrow\textsc{App2}$$

Lema 3.4.3 (Reflexividad)**.** Para todo término t vale que $t \Rightarrow t$.

Demostración. Por inducción en t.

1. $t = x$. Luego, por $\Rightarrow$Var vale $x \Rightarrow x$.
2. $t = \lambda x.\, t'$. Por HI vale $t' \Rightarrow t'$. Luego, por $\Rightarrow$Abs vale $\lambda x.\, t' \Rightarrow \lambda x.\, t'$.
3. $t = s\, u$. Por HI vale $s \Rightarrow s$ y $u \Rightarrow u$. Luego, por $\Rightarrow$App1 vale $s\, u \Rightarrow s\, u$.
4. $t = \bullet\, t'$. Por HI vale $t' \Rightarrow t'$. Luego, por $\Rightarrow$Delay vale $\bullet\, t' \Rightarrow \bullet\, t'$.
5. $t = s[x/u]$. Por HI vale $s \Rightarrow s$ y $u \Rightarrow u$. Luego, por $\Rightarrow$Wait1 vale $s[x/u] \Rightarrow s[x/u]$.

□

Lema 3.4.4 ($\rightarrow \subseteq \Rightarrow$)**.** Si $t \rightarrow t'$ entonces $t \Rightarrow t'$.

Demostración. Por inducción en t.

1. $t = x$. Este caso es imposible pues no existe t' tal que $x \rightarrow t'$.
2. $t = \lambda x.\, s$. Sabemos que $t \rightarrow t'$, es decir, $\lambda x.\, s \rightarrow \lambda x.\, s'$ donde $s \rightarrow s'$. Por HI, $s \Rightarrow s'$. Luego, por $\Rightarrow$Abs, $\lambda x.\, s \Rightarrow \lambda x.\, s'$.
3. $t = s\, u$. Sabemos que $t \rightarrow t'$. Puede ser que la reducción sea interna a s, interna a u o en la raíz.

    3.1 Interna a s, es decir $s\, u \rightarrow s'\, u$ con $s \rightarrow s'$: por HI, $s \Rightarrow s'$ y por Lema 3.4.3, $u \Rightarrow u$. Luego, por $\Rightarrow$App1, vale $s\, u \Rightarrow s'\, u$.

    3.2 Interna a u, es decir $s\, u \rightarrow s\, u'$ con $u \rightarrow u'$: por HI, $u \Rightarrow u'$ y por Lema 3.4.3, $s \Rightarrow s$. Luego, por $\Rightarrow$App1, vale $s\, u \Rightarrow s\, u'$.

3.3 En la raíz, es decir $s = \lambda x.\, r$ y el paso de reducción es de la forma $(\lambda x.\, r)\, u \to_{\beta\bullet} \bullet\, r\{x := u\}$. Por Lema 3.4.3, $r \Rrightarrow r$ y $u \Rrightarrow u$. Luego, por $\Rrightarrow$App2, $(\lambda x.\, r)\, u \Rrightarrow \bullet\, r\{x := u\}$.

4. $t = \bullet\, s$. Sabemos que $t \to t'$, es decir, $\bullet\, s \to \bullet\, s'$ con $s \to s'$. Por HI, $s \Rrightarrow s'$. Luego, por $\Rrightarrow$Delay, $\bullet\, s \Rrightarrow \bullet\, s'$.

5. $t = s[x/u]$. Sabemos que $t \to t'$. Puede ser que la reducción sea interna a s, interna a u o en la raíz.

   5.1 Interna a s, es decir $s[x/u] \to s'[x/u]$ con $s \to s'$: por HI, $s \Rrightarrow s'$ y por Lema 3.4.3, $u \Rrightarrow u$. Luego, por $\Rrightarrow$Wait1, vale $s[x/u] \Rrightarrow s'[x/u]$.

   5.2 Interna a u, es decir $s[x/u] \to s[x/u']$ con $u \to u'$: por HI, $u \Rrightarrow u'$ y por Lema 3.4.3, $s \Rrightarrow s$. Luego, por $\Rrightarrow$Wait1, vale $s[x/u] \Rrightarrow s[x/u']$.

   5.3 En la raíz, es decir, $u = \bullet\, r$ y el paso de reducción es de la forma $s[x/\bullet\, r] \to_w \bullet\, s\{x := r\}$: por Lema 3.4.3, $s \Rrightarrow s$ y $r \Rrightarrow r$. Luego, por $\Rrightarrow$Wait2, $s[x/\bullet\, r] \Rrightarrow \bullet\, s\{x := r\}$.

□

Lema 3.4.5 ($\Rrightarrow \subseteq \twoheadrightarrow$)**.** Si $t \Rrightarrow t'$ entonces $t \twoheadrightarrow t'$.

Demostración. Por inducción en la derivación del juicio $t \Rrightarrow t'$.

1. $\Rrightarrow$Var, $x \Rrightarrow x$. Vale $x \twoheadrightarrow x$ por reflexividad de $\twoheadrightarrow$.

2. $\Rrightarrow$Abs, $\lambda x.\, t \Rrightarrow \lambda x.\, t'$ donde $t \Rrightarrow t'$. Por HI vale que $t \twoheadrightarrow t'$. Luego, $\lambda x.\, t \twoheadrightarrow \lambda x.\, t'$.

3. $\Rrightarrow$App1, $t\, s \Rrightarrow t'\, s'$ donde $t \Rrightarrow t'$ y $s \Rrightarrow s'$. Por HI vale que $t \twoheadrightarrow t'$ y $s \twoheadrightarrow s'$. Luego, $t\, s \twoheadrightarrow t'\, s \twoheadrightarrow t'\, s'$.

4. $\Rrightarrow$App2, $(\lambda x.\, t)\, s \Rrightarrow \bullet\, t'\{x := s'\}$ donde $t \Rrightarrow t'$ y $s \Rrightarrow s'$. Por HI vale que $t \twoheadrightarrow t'$ y $s \twoheadrightarrow s'$. Luego, $(\lambda x.\, t)\, s \twoheadrightarrow (\lambda x.\, t')\, s \twoheadrightarrow (\lambda x.\, t')\, s' \to \bullet\, t'\{x := s'\}$.

5. $\Rrightarrow$Delay, $\bullet\, t \Rrightarrow \bullet\, t'$ donde $t \Rrightarrow t'$. Por HI vale que $t \twoheadrightarrow t'$. Luego, $\bullet\, t \twoheadrightarrow \bullet\, t'$.

6. $\Rrightarrow$Wait1, $t[x/s] \Rrightarrow t'[x/s']$ donde $t \Rrightarrow t'$ y $s \Rrightarrow s'$. Por HI vale que $t \twoheadrightarrow t'$ y $s \twoheadrightarrow s'$. Luego, $t[x/s] \twoheadrightarrow t'[x/s] \twoheadrightarrow t'[x/s']$.

7. $\Rrightarrow$Wait2, $t[x/\bullet\, s] \Rrightarrow \bullet\, t'\{x := s'\}$ donde $t \Rrightarrow t'$ y $s \Rrightarrow s'$. Por HI vale que $t \twoheadrightarrow t'$ y $s \twoheadrightarrow s'$. Luego, $t[x/\bullet\, s] \twoheadrightarrow t'[x/\bullet\, s] \twoheadrightarrow t'[x/\bullet\, s'] \to \bullet\, t'\{x := s'\}$.

□

Lema 3.4.6 (La sustitución conmuta con la reducción en paralelo)**.** Si $t \Rrightarrow t'$ y $s \Rrightarrow s'$ entonces $t\{x := s\} \Rrightarrow t'\{x := s'\}$.

Demostración. Por inducción en la derivación del juicio $t \Rrightarrow t'$.

1. $\Rrightarrow$Var, $t = y \Rrightarrow y = t'$.

   1.1 Si $y \neq x$ entonces, $t\{x := s\} = y \Rrightarrow y = t'\{x := s'\}$ por $\Rrightarrow$Var.

   1.2 Si $y = x$ entonces, $t\{x := s\} = s \Rrightarrow s' = t'\{x := s'\}$ por hipótesis.

2. $\Rightarrow$Abs, $t = (\lambda y.\, u)$. Asumimos por α-equivalencia $x \neq y$ y que $y \notin \mathsf{fv}(s)$. Sabemos $t = \lambda y.\, u \Rightarrow \lambda y.\, u'$ con $u \Rightarrow u'$. Por HI $u\{x := s\} \Rightarrow u'\{x := s'\}$. Por $\Rightarrow$Abs, $\lambda y.\, (u\{x := s\}) \Rightarrow \lambda y.\, (u'\{x := s'\})$. Además, $\lambda y.\, (u\{x := s\}) = (\lambda y.\, u)\{x := s\}$ pues $y \notin \{x\} \cup \mathsf{fv}(s)$ y $\lambda y.\, (u'\{x := s'\}) = (\lambda y.\, u')\{x := s'\}$ pues $y\{x\} \cup \mathsf{fv}(s')$. Observar que como $s \Rightarrow s'$, entonces $s \twoheadrightarrow s'$. Luego, por Lema 3.1.7, $\mathsf{fv}(s) \supseteq \mathsf{fv}(s')$. Por lo tanto, $(\lambda y.\, u)\{x := s\} = (\lambda y.\, u')\{x := s'\}$.

3. $\Rightarrow$App1, $t = u\, r$. Sabemos que $t = u\, r \Rightarrow t'$ se deriva de $\Rightarrow$App1, luego $t' = u'\, r'$ con $u \Rightarrow u'$ y $r \Rightarrow r'$. Por HI $u\{x := s\} \Rightarrow u'\{x := s'\}$ y $r\{x := s\} \Rightarrow r'\{x := s'\}$. Por lo tanto, $t\{x := s\} = u\{x := s\}\, r\{x := s\}$. Por $\Rightarrow$App1, $u\{x := s\}\, r\{x := s\} \Rightarrow u'\{x := s'\}\, r'\{x := s'\} = t'\{x := s'\}$.

4. $\Rightarrow$App2, $t = (\lambda y.\, u)\, r$. Asumimos por α-equivalencia que $y \notin \{x\} \cup \mathsf{fv}(s)$. Sabemos $t = (\lambda y.\, u)\, r \Rightarrow \bullet u'\{y := r'\} = t'$ donde $u \Rightarrow u'$ y $r \Rightarrow r'$. Por HI $u\{x := s\} \Rightarrow u'\{x := s'\}$ y $r\{x := s\} \Rightarrow r'\{x := s'\}$. Por lo tanto,

$$\begin{array}{rcll} t\{x := s\} & = & ((\lambda y.\, u)\, r)\{x := s\} & \\ & = & (\lambda y.\, u\{x := s\})\, r\{x := s\} & \text{pues } y \notin \{x\} \cup \mathsf{fv}(s) \\ & \Rightarrow & \bullet u'\{x := s'\}\{y := r'\{x := s'\}\} & \text{por } \Rightarrow\text{App2} \\ & = & \bullet u'\{y := r'\}\{x := s'\} & \text{pues } y \notin \{x\} \cup \mathsf{fv}(s) \\ & & & \text{y por Lema 3.1.9} \\ & = & t'\{x := s'\} & \end{array}$$

5. $\Rightarrow$Delay, $t = \bullet u$. Sabemos $t = \bullet u \Rightarrow \bullet u' = t'$ con $u \Rightarrow u'$. Por HI $u\{x := s\} \Rightarrow u'\{x := s'\}$. Luego, por $\Rightarrow$Delay, $\bullet u\{x := s\} \Rightarrow \bullet u'\{x := s'\}$.

6. $\Rightarrow$Wait1, $t = u[y/r]$. Asumimos por α-equivalencia que $y \notin \{x\} \cup \mathsf{fv}(s)$. Sabemos $t = u[y/r] \Rightarrow u'[y/r'] = t'$ con $u \Rightarrow u'$ y $r \Rightarrow r'$. Por HI $u\{x := s\} \Rightarrow u'\{x := s'\}$ y $r\{x := s\} \Rightarrow r'\{x := s'\}$. Por $\Rightarrow$Wait1, $(u\{x := s\})[y/(r\{x := s\})] \Rightarrow (u'\{x := s'\})[y/(r'\{x := s'\})]$. Luego, $(u\{x := s\})[y/(r\{x := s\})] = (u[y/r])\{x := s\}$ pues $y \notin \{x\} \cup \mathsf{fv}(s)$. Además, $(u'\{x := s'\})[y/(r'\{x := s'\})] = (u'[y/r'])\{x := s'\}$ pues $y \notin \{x\} \cup \mathsf{fv}(s')$ por Lema 3.1.7. Entonces, $(u[y/r])\{x := s\} \Rightarrow (u'[y/r'])\{x := s'\}$.

7. $\Rightarrow$Wait2, $t = u[y/\bullet r]$. Asumimos por α-equivalencia que $y \notin \{x\} \cup \mathsf{fv}(s)$. Sabemos $t = u[y/\bullet r] \Rightarrow \bullet u'\{y := r'\} = t'$ con $u \Rightarrow u'$ y $r \Rightarrow r'$. Por HI $u\{x := s\} \Rightarrow u'\{x := s'\}$ y $r\{x := s\} \Rightarrow r'\{x := s'\}$. Por lo tanto,

$$\begin{array}{rcll} t\{x := s\} & = & (u[y/\bullet r])\{x := s\} & \\ & = & u\{x := s\}[y/\bullet r\{x := s\}] & \text{pues } y \notin \{x\} \cup \mathsf{fv}(s) \\ & & & \text{y por } \Rightarrow\text{Delay} \\ & \Rightarrow & \bullet u'\{x := s'\}\{y := r'\{x := s'\}\} & \text{por } \Rightarrow\text{Wait2} \\ & = & \bullet u'\{y := r'\}\{x := s'\} & \text{pues } y \notin \{x\} \cup \mathsf{fv}(s) \\ & & & \text{y por Lema 3.1.9} \\ & = & t'\{x := s'\} & \end{array}$$

$\square$

Lema 3.4.7 (Propiedad del diamante)**.** Si $t \Rightarrow t_1$ y $t \Rightarrow t_2$, entonces existe t_3 tal que $t_1 \Rightarrow t_3$ y $t_2 \Rightarrow t_3$.

Demostración. Queremos ver que vale:

$$\begin{tikzcd} t \arrow[r, Rightarrow] \arrow[d, Rightarrow] & t_1 \arrow[d, Rightarrow, dotted] \\ t_2 \arrow[r, Rightarrow, dotted] & t_3 \end{tikzcd}$$

Por inducción en t.

1. $t = x$. Luego, $t \Rightarrow t_1$ y $t \Rightarrow t_2$ solo se pueden haber deducido de $\Rightarrow$VAR. Por lo tanto, $t_1 = t_2 = x$. Entonces tomando $t_3 := x$ vale $t_1 \Rightarrow t_3$ y $t_2 \Rightarrow t_3$.

2. $t = \lambda x.\, s$. Luego $t \Rightarrow t_1$ y $t \Rightarrow t_2$ solo se pueden haber deducido de $\Rightarrow$ABS. La situación es la siguiente:

$$\begin{tikzcd} t = \lambda x.\, s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \lambda x.\, s_1 = t_1 \\ t_2 = \lambda x.\, s_2 & \end{tikzcd} \quad \text{donde} \quad \begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \\ s_2 & \end{tikzcd}$$

Por HI, existe s_3 tal que $s_1 \Rightarrow s_3$ y $s_2 \Rightarrow s_3$.

$$\begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \arrow[d, Rightarrow, dotted] \\ s_2 \arrow[r, Rightarrow, dotted] & s_3 \end{tikzcd}$$

Luego por $\Rightarrow$ABS vale, $\lambda x.\, s_1 \Rightarrow \lambda x.\, s_3$ y $\lambda x.\, s_2 \Rightarrow \lambda x.\, s_3$. Basta tomar $t_3 := \lambda x.\, s_3$.

$$\begin{tikzcd} t = \lambda x.\, s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \lambda x.\, s_1 = t_1 \arrow[d, Rightarrow, dotted] \\ t_2 = \lambda x.\, s_2 \arrow[r, Rightarrow, dotted] & \lambda x.\, s_3 = t_3 \end{tikzcd}$$

3. $t = s\, u$. Luego $t \Rightarrow t_1$ y $t \Rightarrow t_2$ podrían reducir cada una de dos formas, dependiendo si se concluye por $\Rightarrow$APP1 o por $\Rightarrow$APP2. Por lo tanto, hay cuatro casos. Sólo consideramos tres casos pues $\Rightarrow$APP1/$\Rightarrow$APP2 y $\Rightarrow$APP2/$\Rightarrow$APP1 son simétricos.

   3.1 $\Rightarrow$APP1/$\Rightarrow$APP1: luego, $t_1 = s_1\, u_1$ con $s \Rightarrow s_1$ y $u \Rightarrow u_1$ y, $t_2 = s_2\, u_2$ con $s \Rightarrow s_2$ y $u \Rightarrow u_2$. La situación es la siguiente:

$$\begin{tikzcd} t = s\, u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1\, u_1 = t_1 \\ t_2 = s_2\, u_2 & \end{tikzcd} \quad \text{donde} \quad \begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \\ s_2 & \end{tikzcd} \text{ y } \begin{tikzcd} u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \\ u_2 & \end{tikzcd}$$

   Por HI, existen s_3 y u_3 tales que $s_1 \Rightarrow s_3$ y $s_2 \Rightarrow s_3$ y, $u_1 \Rightarrow u_3$ y $u_2 \Rightarrow u_3$.

$$\begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \arrow[d, Rightarrow, dotted] \\ s_2 \arrow[r, Rightarrow, dotted] & s_3 \end{tikzcd} \text{ y } \begin{tikzcd} u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \arrow[d, Rightarrow, dotted] \\ u_2 \arrow[r, Rightarrow, dotted] & u_3 \end{tikzcd}$$

Luego por $\Rightarrow$App1 vale, $s_1\, u_1 \Rightarrow s_3\, u_3$ y $s_2\, u_2 \Rightarrow s_3\, u_3$. Basta tomar $t_3 := s_3\, u_3$.

$$\begin{tikzcd}
t = s\, u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1\, u_1 = t_1 \arrow[d, dotted, Rightarrow] \\
t_2 = s_2\, u_2 \arrow[r, dotted, Rightarrow] & s_3\, u_3 = t_3
\end{tikzcd}$$

3.2 $\Rightarrow$App1/$\Rightarrow$App2: luego, $s = (\lambda x.\, r)$, por lo tanto, $t = (\lambda x.\, r)\, u$. Entonces, $t_1 = s_1\, u_1$ con $s = (\lambda x.\, r) \Rightarrow s_1$ y $u \Rightarrow u_1$. Observar que $s = (\lambda x.\, r) \Rightarrow s_1$ necesariamente se deriva por $\Rightarrow$Abs de $r \Rightarrow r_1$ y $s_1 = \lambda x.\, r_1$. Además, $t_2 = \bullet\, r_2\{x := u_2\}$ con $r \Rightarrow r_2$ y $u \Rightarrow u_2$. La situación es la siguiente:

$$\begin{tikzcd}
t = (\lambda x.\, r)\, u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & (\lambda x.\, r_1)\, u_1 = t_1 & \text{donde} & r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 & \text{y} & u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \\
t_2 = \bullet\, r_2\{x := u_2\} & & & r_2 & & & u_2 &
\end{tikzcd}$$

Por HI, existen r_3 y u_3 tales que $r_1 \Rightarrow r_3$ y $r_2 \Rightarrow r_3$ y, $u_1 \Rightarrow u_3$ y $u_2 \Rightarrow u_3$.

$$\begin{tikzcd}
r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 \arrow[d, dotted, Rightarrow] & \text{y} & u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \arrow[d, dotted, Rightarrow] \\
r_2 \arrow[r, dotted, Rightarrow] & r_3 & & u_2 \arrow[r, dotted, Rightarrow] & u_3
\end{tikzcd}$$

Entonces, por $\Rightarrow$App2 $t_1 = (\lambda x.\, r_1)\, u_1 \Rightarrow \bullet\, r_3\{x := u_3\}$. Por Lema 3.4.6, $r_2\{x := u_2\} \Rightarrow r_3\{x := u_3\}$ y por $\Rightarrow$Delay, $t_2 = \bullet\, r_2\{x := u_2\} \Rightarrow \bullet\, r_3\{x := u_3\}$. Basta tomar $t_3 := \bullet\, r_3\{x := u_3\}$.

$$\begin{tikzcd}
t = (\lambda x.\, r)\, u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & (\lambda x.\, r_1)\, u_1 = t_1 \arrow[d, dotted, Rightarrow] \\
t_2 = \bullet\, r_2\{x := u_2\} \arrow[r, dotted, Rightarrow] & \bullet\, r_3\{x := u_3\} = t_3
\end{tikzcd}$$

3.3 $\Rightarrow$App2/$\Rightarrow$App2: luego $t = (\lambda x.\, r)\, u$. Entonces, $t_1 = \bullet\, r_1\{x := u_1\}$ con $r \Rightarrow r_1$ y $u \Rightarrow u_1$, y $t_2 = \bullet\, r_2\{x := u_2\}$ con $r \Rightarrow r_2$ y $u \Rightarrow u_2$. La situación es la siguiente:

$$\begin{tikzcd}
t = (\lambda x.\, r)\, u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \bullet\, r_1\{x := u_1\} = t_1 & \text{donde} & r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 & \text{y} & u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \\
t_2 = \bullet\, r_2\{x := u_2\} & & & r_2 & & & u_2 &
\end{tikzcd}$$

Por HI, existen r_3 y u_3 tales que $r_1 \Rightarrow r_3$ y $r_2 \Rightarrow r_3$ y, $u_1 \Rightarrow u_3$ y $u_2 \Rightarrow u_3$.

$$\begin{tikzcd}
r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 \arrow[d, dotted, Rightarrow] & \text{y} & u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \arrow[d, dotted, Rightarrow] \\
r_2 \arrow[r, dotted, Rightarrow] & r_3 & & u_2 \arrow[r, dotted, Rightarrow] & u_3
\end{tikzcd}$$

Por Lema 3.4.6 y $\Rightarrow$Delay vale, $\bullet\, r_1\{x := u_1\} \Rightarrow \bullet\, r_3\{x := u_3\}$ y $\bullet\, r_2\{x := u_2\} \Rightarrow \bullet\, r_3\{x := u_3\}$. Basta tomar $t_3 := \bullet\, r_3\{x := u_3\}$.

$$\begin{tikzcd} t = (\lambda x.\, r)\, u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \bullet\, r_1\{x := u_1\} = t_1 \arrow[d, dotted, Rightarrow] \\ t_2 = \bullet\, r_2\{x := u_2\} \arrow[r, dotted, Rightarrow] & \bullet\, r_3\{x := u_3\} = t_3 \end{tikzcd}$$

4. $t = \bullet\, s$. Luego $t \Rightarrow t_1$ y $t \Rightarrow t_2$ solo se pueden haber deducido de $\Rightarrow$Delay. La situación es la siguiente:

$$\begin{tikzcd} t = \bullet\, s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \bullet\, s_1 = t_1 \\ t_2 = \bullet\, s_2 & \end{tikzcd} \quad \text{donde} \quad \begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \\ s_2 & \end{tikzcd}$$

Por HI, existe s_3 tal que $s_1 \Rightarrow s_3$ y $s_2 \Rightarrow s_3$.

$$\begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \arrow[d, dotted, Rightarrow] \\ s_2 \arrow[r, dotted, Rightarrow] & s_3 \end{tikzcd}$$

Luego por $\Rightarrow$Delay vale, $\bullet\, s_1 \Rightarrow \bullet\, s_3$ y $\bullet\, s_2 \Rightarrow \bullet\, s_3$. Basta tomar $t_3 := \bullet\, s_3$.

$$\begin{tikzcd} t = \bullet\, s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \bullet\, s_1 = t_1 \arrow[d, dotted, Rightarrow] \\ t_2 = \bullet\, s_2 \arrow[r, dotted, Rightarrow] & \bullet\, s_3 = t_3 \end{tikzcd}$$

5. $t = s[x/u]$. Luego $t \Rightarrow t_1$ y $t \Rightarrow t_2$ podrían reducir cada una de dos formas, dependiendo si se concluye por $\Rightarrow$Wait1 o por $\Rightarrow$Wait2. Por lo tanto, hay cuatro casos. Sólo consideramos tres casos pues $\Rightarrow$Wait1/$\Rightarrow$Wait2 y $\Rightarrow$Wait2/$\Rightarrow$Wait1 son simétricos.

   5.1 $\Rightarrow$Wait1/$\Rightarrow$Wait1: luego, $t_1 = s_1[x/u_1]$ con $s \Rightarrow s_1$ y $u \Rightarrow u_1$ y, $t_2 = s_2[x/u_2]$ con $s \Rightarrow s_2$ y $u \Rightarrow u_2$. La situación es la siguiente:

$$\begin{tikzcd} t = s[x/u] \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1[x/u_1] = t_1 \\ t_2 = s_2[x/u_2] & \end{tikzcd} \quad \text{donde} \quad \begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \\ s_2 & \end{tikzcd} \quad \text{y} \quad \begin{tikzcd} u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \\ u_2 & \end{tikzcd}$$

   Por HI, existen s_3 y u_3 tales que $s_1 \Rightarrow s_3$ y $s_2 \Rightarrow s_3$ y, $u_1 \Rightarrow u_3$ y $u_2 \Rightarrow u_3$.

$$\begin{tikzcd} s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \arrow[d, dotted, Rightarrow] \\ s_2 \arrow[r, dotted, Rightarrow] & s_3 \end{tikzcd} \quad \text{y} \quad \begin{tikzcd} u \arrow[r, Rightarrow] \arrow[d, Rightarrow] & u_1 \arrow[d, dotted, Rightarrow] \\ u_2 \arrow[r, dotted, Rightarrow] & u_3 \end{tikzcd}$$

Luego por $\Rightarrow$Wait1 vale, $s_1[x/u_1] \Rightarrow s_3[x/u_3]$ y $s_2[x/u_2] \Rightarrow s_3[x/u_3]$. Basta tomar $t_3 := s_3[x/u_3]$.

$$\begin{tikzcd} t = s[x/u] \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1[x/u_1] = t_1 \arrow[d, Rightarrow, dotted] \\ t_2 = s_2[x/u_2] \arrow[r, Rightarrow, dotted] & s_3[x/u_3] = t_3 \end{tikzcd}$$

5.2 $\Rightarrow$Wait1/$\Rightarrow$Wait2: luego, $u = \bullet\, r$, por lo tanto, $t = s[x/\bullet\, r]$. Entonces, $t_1 = s_1[x/u_1]$ con $u = \bullet\, r \Rightarrow u_1$ y $s \Rightarrow s_1$. Observar que $u = \bullet\, r \Rightarrow u_1$ necesariamente se deriva por $\Rightarrow$Delay de $r \Rightarrow r_1$ y $u_1 = \bullet\, r_1$. Además, $t_2 = \bullet\, s_2\{x := r_2\}$ con $r \Rightarrow r_2$ y $s \Rightarrow s_2$. La situación es la siguiente:

$$\begin{tikzcd} t = s[x/\bullet\, r] \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1[x/\bullet\, r_1] = t_1 \ \text{donde} & r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 \ \text{y} & s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \\ t_2 = \bullet\, s_2\{x := r_2\} & & r_2 & & s_2 & \end{tikzcd}$$

Por HI, existen r_3 y s_3 tales que $r_1 \Rightarrow r_3$ y $r_2 \Rightarrow r_3$ y, $s_1 \Rightarrow s_3$ y $s_2 \Rightarrow s_3$.

$$\begin{tikzcd} r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 \ \text{y} \arrow[d, Rightarrow, dotted] & s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \arrow[d, Rightarrow, dotted] \\ r_2 \arrow[r, Rightarrow, dotted] & r_3 & s_2 \arrow[r, Rightarrow, dotted] & s_3 \end{tikzcd}$$

Entonces, por $\Rightarrow$Wait2 $t_1 = s_1[x/\bullet\, r_1] \Rightarrow \bullet\, s_3\{x := r_3\}$. Por Lema 3.4.6, $s_2\{x := r_2\} \Rightarrow s_3\{x := r_3\}$ y por $\Rightarrow$Delay, $t_2 = \bullet\, s_2\{x := r_2\} \Rightarrow \bullet\, s_3\{x := r_3\}$. Basta tomar $t_3 := \bullet\, s_3\{x := r_3\}$.

$$\begin{tikzcd} t = s[x/\bullet\, r] \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1[x/\bullet\, r_1] = t_1 \arrow[d, Rightarrow, dotted] \\ t_2 = \bullet\, s_2\{x := r_2\} \arrow[r, Rightarrow, dotted] & \bullet\, s_3\{x := r_3\} = t_3 \end{tikzcd}$$

5.3 $\Rightarrow$Wait2/$\Rightarrow$Wait2: luego $t = s[x/\bullet\, r]$. Entonces, $t_1 = \bullet\, s_1\{x := r_1\}$ con $r \Rightarrow r_1$ y $s \Rightarrow s_1$, y $t_2 = \bullet\, s_2\{x := r_2\}$ con $r \Rightarrow r_2$ y $s \Rightarrow s_2$. La situación es la siguiente:

$$\begin{tikzcd} t = s[x/\bullet\, r] \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \bullet\, s_1\{x := r_1\} = t_1 \ \text{donde} & r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 \ \text{y} & s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \\ t_2 = \bullet\, s_2\{x := r_2\} & & r_2 & & s_2 & \end{tikzcd}$$

Por HI, existen r_3 y s_3 tales que $r_1 \Rightarrow r_3$ y $r_2 \Rightarrow r_3$ y, $s_1 \Rightarrow s_3$ y $s_2 \Rightarrow s_3$.

$$\begin{tikzcd} r \arrow[r, Rightarrow] \arrow[d, Rightarrow] & r_1 \ \text{y} \arrow[d, Rightarrow, dotted] & s \arrow[r, Rightarrow] \arrow[d, Rightarrow] & s_1 \arrow[d, Rightarrow, dotted] \\ r_2 \arrow[r, Rightarrow, dotted] & r_3 & s_2 \arrow[r, Rightarrow, dotted] & s_3 \end{tikzcd}$$

Por Lema 3.4.6 y $\Rightarrow$Delay vale, $\bullet\, s_1\{x := r_1\} \Rightarrow \bullet\, s_3\{x := r_3\}$ y $\bullet\, s_2\{x := r_2\} \Rightarrow \bullet\, s_3\{x := r_3\}$. Basta tomar $t_3 := \bullet\, s_3\{x := r_3\}$.

$$\begin{tikzcd}
t = s[x/\bullet\, r] \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \bullet\, s_1\{x := r_1\} = t_1 \arrow[d, Rightarrow, dotted] \\
t_2 = \bullet\, s_2\{x := r_2\} \arrow[r, Rightarrow, dotted] & \bullet\, s_3\{x := r_3\} = t_3
\end{tikzcd}$$

□

Teorema 3.4.8. El cálculo-$\lambda^\bullet$ es confluente.

Demostración. Por Lemas 3.4.4, 3.4.5 y 3.4.7, usando el argumento de Tait–Martin-Löf (Teorema 2.1.5). □

4. TERMINACIÓN

En este capítulo estudiamos propiedades de terminación del cálculo-$\lambda^\bullet$. Abordamos el problema usando dos técnicas radicalmente distintas.

En primer lugar, en la Sección 4.1, probamos que el cálculo-$\lambda^\bullet$ tipado es fuertemente normalizante dando una traducción que convierte cada término tipable del cálculo-$\lambda^\bullet$ en un término tipable del cálculo-λ, y basándonos en el hecho de que este último es fuertemente normalizante.

En segundo lugar, en la sección Sección 4.2, aprovechamos el hecho de que el cálculo-$\lambda^\bullet$ tiene marcadores del paso del tiempo para dar cotas explícitas para la longitud de ciertas reducciones a forma normal, usando una propiedad del cálculo-$\lambda^\bullet$ tipado a la que llamamos "productividad". Esto da una demostración directa de terminación débil de términos tipables (bajo hipótesis apropiadas). La idea es que un término de tipo α debe estar *listo*, es decir, "inmediatamente disponible" sin necesidad de esperar una unidad de tiempo. La necesidad de esperar una unidad de tiempo para poder acceder al resultado de un cómputo se ve reflejada necesariamente en el tipo. En particular, esto nos permite probar resultados de terminación mediante cotas que se expresan directamente a través de operaciones aritméticas, algo que no es tan sencillo en el cálculo-λ.

4.1. Terminación fuerte por traducción

En esta sección definimos una traducción (Definición 4.1.2) que a cada término t del cálculo-$\lambda^\bullet$ le asocia un término $[\![t]\!]$ del cálculo-λ. Demostramos que la traducción *preserva la tipabilidad* (Lema 4.1.3), es decir, si t es tipable, $[\![t]\!]$ también lo es. Demostramos además que podemos simular las reducciones del cálculo-$\lambda^\bullet$ a través de la traducción (Lema 4.1.5), es decir, que cada paso $t \to s$ tiene asociada una reducción $[\![t]\!] \to^+ [\![s]\!]$. A partir de esto, usando el hecho de que el cálculo-λ es fuertemente normalizante, concluimos que este nuevo cálculo también lo es (Teorema 4.1.6).

Comenzamos dando la definición precisa del fragmento tipado del cálculo-$\lambda^\bullet$:

Definición 4.1.1 (Cálculo-$\lambda^\bullet$ simplemente tipado)**.** El conjunto de *términos tipables* del cálculo-$\lambda^\bullet$ se define de la siguiente manera:

$$\Lambda_\bullet^\to \stackrel{\text{def}}{=} \{t \in \Lambda_\bullet \mid \exists \Gamma, A.\ \Gamma \rhd t : A\}$$

El *cálculo-$\lambda^\bullet$ simplemente tipado* es el sistema de reescritura definido sobre el conjunto de términos $\Lambda_\bullet^\to$ con la relación de reescritura $\to$ ya definida. Observemos que los términos tipables son cerrados por reducción, es decir, si $t \in \Lambda_\bullet^\to$ y $t \to s$ entonces $s \in \Lambda_\bullet^\to$, por el resultado de preservación de tipos (Proposición 3.3.3).

A continuación, demostramos que el cálculo-$\lambda^\bullet$ simplemente tipado es fuertemente normalizante a través de una traducción al cálculo-λ simplemente tipado.

Definición 4.1.2 (Traducción "olvidadiza")**.** Definimos una traducción $[\![-]\!]$ que dado un tipo (o, respectivamente un contexto de tipado o un término) del cálculo-$\lambda^\bullet$ devuelve un tipo (o, respectivamente un contexto de tipado o un término) del cálculo-λ.

1. Sobre los tipos:

$$\begin{array}{lcl} [\![\alpha]\!] & \overset{\text{def}}{=} & \alpha \\ [\![A \to B]\!] & \overset{\text{def}}{=} & [\![A]\!] \to [\![B]\!] \\ [\![\bullet\, A]\!] & \overset{\text{def}}{=} & [\![A]\!] \end{array}$$

2. Sobre los contextos de tipado:

$$[\![x_1 : A_1, \ldots, x_n : A_n]\!] \overset{\text{def}}{=} x_1 : [\![A_1]\!], \ldots, x_n : [\![A_n]\!]$$

3. Sobre los términos:

$$\begin{array}{lcl} [\![x]\!] & \overset{\text{def}}{=} & x \\ [\![\lambda x.\, t]\!] & \overset{\text{def}}{=} & \lambda x.\, [\![t]\!] \\ [\![t\, s]\!] & \overset{\text{def}}{=} & [\![t]\!]\, [\![s]\!] \\ [\![\bullet\, t]\!] & \overset{\text{def}}{=} & [\![t]\!] \\ [\![t[x/s]]\!] & \overset{\text{def}}{=} & (\lambda x.\, [\![t]\!])\, [\![s]\!] \end{array}$$

Lema 4.1.3 (La traducción preserva la tipabilidad)**.** Si $\Gamma \rhd t : A$ en $\Lambda^{\to}_{\bullet}$ entonces $[\![\Gamma]\!] \vdash [\![t]\!] : [\![A]\!]$.

Demostración. Por inducción en la derivación del juicio $\Gamma \rhd t : A$.

1. $\mathrm{T}^{\bullet}\textsc{Var}$, $t = x$:

$$\text{Sabemos que } \frac{}{\Gamma, x : A \rhd x : A}\ \mathrm{T}^{\bullet}\textsc{Var} \text{ entonces, } \frac{}{[\![\Gamma]\!], x : [\![A]\!] \vdash x : [\![A]\!]}\ \mathrm{T}\textsc{Var}$$

2. $\mathrm{T}^{\bullet}\textsc{Abs}$, $t = \lambda y.\, s$:

$$\text{Sabemos que } \frac{\Gamma, x : A \rhd s : B}{\Gamma \rhd \lambda x.\, s : A \to B}\ \mathrm{T}^{\bullet}\textsc{Abs}$$

$$\text{entonces, } \frac{\dfrac{\text{HI}}{[\![\Gamma]\!], x : [\![A]\!] \vdash [\![s]\!] : [\![B]\!]}}{[\![\Gamma]\!] \vdash \lambda x.\, [\![s]\!] : [\![A]\!] \to [\![B]\!]}\ \mathrm{T}\textsc{Abs}$$

3. $\mathrm{T}^{\bullet}\textsc{App}$, $t = s\, u$:

$$\text{Sabemos que } \frac{\Gamma \rhd s : A \to B \quad \Gamma \rhd u : A}{\Gamma \rhd s\, u : \bullet\, B}\ \mathrm{T}^{\bullet}\textsc{App}$$

$$\text{entonces, } \frac{\dfrac{\text{HI}}{[\![\Gamma]\!] \vdash [\![s]\!] : [\![A]\!] \to [\![B]\!]} \qquad \dfrac{\text{HI}}{[\![\Gamma]\!] \vdash [\![u]\!] : [\![A]\!]}}{[\![\Gamma]\!] \vdash [\![s]\!]\, [\![u]\!] : [\![B]\!]}\ \mathrm{T}\textsc{App}$$

4. $\mathrm{T}^{\bullet}\textsc{Delay}$, $t = \bullet\, s$:

$$\text{Sabemos que } \frac{\Gamma \rhd s : A}{\Gamma \rhd \bullet\, s : \bullet\, A}\ \mathrm{T}^{\bullet}\textsc{Delay} \text{ y por HI vale } [\![\Gamma]\!] \vdash [\![s]\!] : [\![A]\!] \text{ pues } [\![A]\!] = [\![\bullet\, A]\!]$$

5. $T^{\bullet}$Wait, $t = s[y/u]$:

$$\text{Sabemos que } \frac{\Gamma, x : A \rhd s : B \qquad \Gamma \rhd u : \bullet A}{\Gamma \rhd s[x/u] : \bullet B}\ \mathrm{T^{\bullet}W\small{AIT}} \text{ entonces,}$$

$$\dfrac{\dfrac{\dfrac{\text{HI}}{[\![\Gamma]\!], x : [\![A]\!] \vdash [\![s]\!] : [\![B]\!]}}{[\![\Gamma]\!] \vdash \lambda x.\, [\![s]\!] : [\![A]\!] \to [\![B]\!]}\ \mathrm{TA{\small BS}} \qquad \dfrac{\text{HI}}{[\![\Gamma]\!] \vdash [\![u]\!] : [\![A]\!]}}{[\![\Gamma]\!] \vdash (\lambda x.\, [\![s]\!])\, [\![u]\!] : [\![B]\!]}\ \mathrm{TA{\small PP}}$$

$\square$

Lema 4.1.4 (La traducción conmuta con la sustitución)**.** $[\![t\{x := s\}]\!] = [\![t]\!]\{x := [\![s]\!]\}$

Demostración. Por inducción en t.

1. $t = y$:

   1.1 $y = x$, entonces,

$$\begin{array}{rcll} [\![t\{x := s\}]\!] & = & [\![x\{x := s\}]\!] & \\ & = & [\![s]\!] & \\ & = & x\{x := [\![s]\!]\} & \\ & = & [\![x]\!]\{x := [\![s]\!]\} & \text{pues } [\![x]\!] = x. \\ & = & [\![t]\!]\{x := [\![s]\!]\} & \end{array}$$

   1.2 $y \neq x$, entonces,

$$\begin{array}{rcl} [\![t\{x := s\}]\!] & = & [\![y\{x := s\}]\!] \\ & = & [\![y]\!] \\ & = & [\![y]\!]\{x := s\} \\ & = & [\![y]\!]\{x := [\![s]\!]\} \\ & = & [\![t]\!]\{x := [\![s]\!]\} \end{array}$$

2. $t = \lambda y.\, u$: asumimos por α-equivalencia que $y \neq x$. Luego,

$$\begin{array}{rcll} [\![t\{x := s\}]\!] & = & [\![(\lambda y.\, u)\{x := s\}]\!] & \\ & = & \lambda y.\, [\![u\{x := s\}]\!] & \\ & = & \lambda y.\, ([\![u]\!]\{x := [\![s]\!]\}) & \text{por HI} \\ & = & [\![(\lambda y.\, u)]\!]\{x := [\![s]\!]\} & \\ & = & [\![t]\!]\{x := [\![s]\!]\} & \end{array}$$

3. $t = u\, r$: Luego,

$$\begin{array}{rcll} [\![t\{x := s\}]\!] & = & [\![(u\, r)\{x := s\}]\!] & \\ & = & [\![u\{x := s\}]\!]\, [\![r\{x := s\}]\!] & \\ & = & ([\![u]\!]\{x := [\![s]\!]\})\, ([\![r]\!]\{x := [\![s]\!]\}) & \text{por HI} \\ & = & [\![u]\!]\, [\![r]\!]\{x := [\![s]\!]\} & \\ & = & [\![u\, r]\!]\{x := [\![s]\!]\} & \\ & = & [\![t]\!]\{x := [\![s]\!]\} & \end{array}$$

4. $t = \bullet\, u$: Luego,

$$\begin{array}{rcll} [\![t\{x := s\}]\!] & = & [\![(\bullet\, u)\{x := s\}]\!] & \\ & = & [\![u\{x := s\}]\!] & \\ & = & [\![u]\!]\{x := [\![s]\!]\} & \text{por HI} \\ & = & [\![\bullet\, u]\!]\{x := [\![s]\!]\} & \\ & = & [\![t]\!]\{x := [\![s]\!]\} & \end{array}$$

5. $t = u[y/r]$: asumimos por α-equivalencia que $y \neq x$. Luego,

$$\begin{array}{rcll} [\![t\{x := s\}]\!] & = & [\![(u[y/r])\{x := s\}]\!] & \\ & = & (\lambda y.\, [\![u\{x := s\}]\!])\, [\![r\{x := s\}]\!] & \\ & = & (\lambda y.\, [\![u]\!]\{x := [\![s]\!]\})\, [\![r]\!]\{x := [\![s]\!]\} & \text{por HI} \\ & = & ((\lambda y.\, [\![u]\!])\, [\![r]\!])\{x := [\![s]\!]\} & \\ & = & [\![u[y/r]]\!]\{x := [\![s]\!]\} & \\ & = & [\![t]\!]\{x := [\![s]\!]\} & \end{array}$$

□

Lema 4.1.5 (Simulación)**.** Si $t \to t'$ entonces $[\![t]\!] \to^+ [\![t']\!]$.

Demostración. Por inducción en t.

1. $t = x$: imposible pues x no reduce.

2. $t = \lambda x.\, s \to \lambda x.\, s' = t'$ con $s \to s'$: por HI $[\![s]\!] \to^+ [\![s']\!]$. Luego,

$$[\![t]\!] = [\![\lambda x.\, s]\!] = \lambda x.\, [\![s]\!] \to^+ \lambda x.\, [\![s']\!] = [\![\lambda x.\, s']\!] = [\![t']\!]$$

3. $t = s\, u$: hay tres subcasos, dependiendo de si la reducción es interna a s, si es interna a u o en la raíz.

   3.1 Interna a s: $t = s\, u \to s'\, u$ con $s \to s'$. Por HI $[\![s]\!] \to^+ [\![s']\!]$. Luego,

   $$[\![t]\!] = [\![s\, u]\!] = [\![s]\!]\, [\![u]\!] \to^+ [\![s']\!]\, [\![u]\!] = [\![s'\, u]\!] = [\![t']\!]$$

   3.2 Interna a u: $t = s\, u \to s\, u'$ con $u \to u'$. Por HI $[\![u]\!] \to^+ [\![u']\!]$. Luego,

   $$[\![t]\!] = [\![s\, u]\!] = [\![s]\!]\, [\![u]\!] \to^+ [\![s]\!]\, [\![u']\!] = [\![s\, u']\!] = [\![t']\!]$$

   3.3 En la raíz: $t = (\lambda x.\, s)\, u \to \bullet\, s\{x := u\} = t'$. Luego,

   $$\begin{array}{rcll} [\![t]\!] & = & [\![(\lambda x.\, s)\, u]\!] & \\ & = & (\lambda x.\, [\![s]\!])\, [\![u]\!] & \\ & \to & [\![s]\!]\{x := [\![u]\!]\} & \\ & = & [\![s\{x := u\}]\!] & \text{por Lema 4.1.4} \\ & = & [\![\bullet\, s\{x := u\}]\!] & \\ & = & [\![t']\!] & \end{array}$$

4. $t = \bullet\, s \to \bullet\, s' = t'$ con $s \to s'$: Por HI $[\![s]\!] \to^+ [\![s']\!]$. Luego,

$$[\![t]\!] = [\![\bullet\, s]\!] \to^+ [\![\bullet\, s']\!] = [\![t']\!]$$

5. $t = s[x/u]$: hay tres subcasos, dependiendo de si la reducción es interna a s, si es interna a u o en la raíz.

   5.1 Interna a s: $t = s[x/u] \to s'[x/u]$ con $s \to s'$. Por HI $[\![s]\!] \to^+ [\![s']\!]$. Luego,

$$[\![t]\!] = [\![s[x/u]]\!] = [\![s]\!][x/[\![u]\!]] \to^+ [\![s']\!][x/[\![u]\!]] = [\![s'[x/u]]\!] = [\![t']\!]$$

   5.2 Interna a u: $t = s[x/u] \to s[x/u']$ con $u \to u'$. Por HI $[\![u]\!] \to^+ [\![u']\!]$. Luego,

$$[\![t]\!] = [\![s[x/u]]\!] = [\![s]\!][x/[\![u]\!]] \to^+ [\![s]\!][x/[\![u']\!]] = [\![s[x/u']]\!] = [\![t']\!]$$

   5.3 En la raíz: $t = s[x/\bullet\, u] \to \bullet\, s\{x := u\} = t'$. Luego,

$$\begin{array}{rcll} [\![t]\!] & = & [\![s[x/\bullet\, u]]\!] & \\ & = & (\lambda x. [\![s]\!])\, [\![\bullet\, u]\!] & \\ & = & (\lambda x. [\![s]\!])\, [\![u]\!] & \\ & \to & [\![s]\!]\{x := [\![u]\!]\} & \\ & = & [\![s\{x := u\}]\!] & \text{por Lema 4.1.4} \\ & = & [\![\bullet\, s\{x := u\}]\!] & \\ & = & [\![t']\!] & \end{array}$$

□

A partir de la traducción $[\![-]\!]$ de la Definición 4.1.2 y usando el hecho de que el cálculo-λ simplemente tipado es SN, podemos concluir que el cálculo-$\lambda^\bullet$ simplemente tipado también es SN. En efecto:

Teorema 4.1.6. El cálculo-$\lambda^\bullet$ simplemente tipado es fuertemente normalizante.

Demostración. Sea $t \in \Lambda_\bullet^\to$. Veamos que no puede existir una secuencia de reducción infinita que empiece en t. Supongamos que existiera, es decir, sean $t_0, t_1, t_2, ... \in \Lambda_\bullet^\to$ tales que:

$$t = t_0 \to t_1 \to t_2 \to ...$$

Por el resultado de simulación (Lema 4.1.5), tenemos que:

$$[\![t]\!] = [\![t_0]\!] \to^+ [\![t_1]\!] \to^+ [\![t_2]\!] \to^+ ...$$

Observemos que para cada $i \geq 0$ el término $t_i \in \Lambda_\bullet^\to$ es tipable, es decir que existen un contexto de tipado Γ_i y un tipo A_i tales que $\Gamma_i \rhd t_i : A_i$. Como la traducción preserva la tipabilidad (Lema 4.1.3) tenemos que $[\![\Gamma_i]\!] \vdash [\![t_i]\!] : [\![A_i]\!]$ en cálculo-λ simplemente tipado, es decir, para cada $i \geq 0$ se tiene que $[\![t_i]\!] \in \Lambda^\to$ es un término tipable del cálculo-λ. Esto implicaría que el cálculo-λ simplemente tipado no es fuertemente normalizante. Esto es absurdo, que provino de suponer que $\Lambda_\bullet^\to$ no era fuertemente normalizante. □

4.2. Terminación débil con cota explícita

En esta sección demostramos que los términos tipables (bajo hipótesis apropiadas) son débilmente normalizantes. Observemos, en primer lugar, que el hecho de que dichos términos son débilmente normalizantes es una consecuencia del hecho de que *todos* los términos tipables del cálculo-$\lambda^\bullet$ son fuertemente normalizantes, como ya fue visto en la sección anterior (Teorema 4.1.6).

Sin embargo, lo interesante es que la demostración que damos en esta sección aprovecha las características del sistema de tipos para proveer una cota explícita de la longitud de la reducción a forma normal. Para esto primero demostramos que después de un paso de reducción el tamaño de un término puede crecer a lo sumo cuadráticamente (Lema 4.2.7), y generalizamos este resultado para n pasos de reducción (Lema 4.2.8). Además, demostramos el resultado de *productividad en un paso* (Lema 4.2.10), que afirma esencialmente que un término de tipo $\bullet A$ se puede reducir hasta "producir" un $\bullet$ en la raíz, en un número de pasos que depende linealmente del tamaño del término. Finalmente, definimos explícitamente una función (Definición 4.2.14) que calcula una cota para el número de pasos de reducción que se necesitan para producir k $\bullet$s en la raíz, en función del tamaño del término (Teorema 4.2.16).

Definición 4.2.1 (Tamaño de un término)**.** Dado un término t definimos su *tamaño* $|t| \in \mathbb{N}_0$ así:

$$\begin{array}{lcl} |x| & \stackrel{\text{def}}{=} & 1 \\ |\lambda x.\, t| & \stackrel{\text{def}}{=} & 1 + |t| \\ |t\, s| & \stackrel{\text{def}}{=} & 1 + |t| + |s| \\ |\bullet\, t| & \stackrel{\text{def}}{=} & 1 + |t| \\ |t[x/s]| & \stackrel{\text{def}}{=} & 1 + |t| + |s| \end{array}$$

Definición 4.2.2 (Número de ocurrencias de una variable)**.** Notamos $\#_x(t)$ al número de ocurrencias libres de x en t. Más precisamente:

$$\begin{array}{lcll} \#_x(y) & \stackrel{\text{def}}{=} & \begin{cases} 1 & \text{si } y = x \\ 0 & \text{si no} \end{cases} & \\ \#_x(\lambda y.\, t) & \stackrel{\text{def}}{=} & \#_x(t) & \text{asumiendo } x \neq y \text{ por } \alpha\text{-equivalencia} \\ \#_x(t\, s) & \stackrel{\text{def}}{=} & \#_x(t) + \#_x(s) & \\ \#_x(\bullet\, t) & \stackrel{\text{def}}{=} & \#_x(t) & \\ \#_x(t[y/s]) & \stackrel{\text{def}}{=} & \#_x(t) + \#_x(s) & \text{asumiendo } x \neq y \text{ por } \alpha\text{-equivalencia} \end{array}$$

Definición 4.2.3 (Término listo)**.** Decimos que t está *listo* sii t no es de la forma $\bullet\, s$.

Observación 4.2.4. Si $\Gamma \rhd t : A \to B$ entonces t es una abstracción, es decir, es de la forma $t = \lambda x.\, t'$.

Observación 4.2.5. $\#_x(t) \leq |t|$

Lema 4.2.6 (Tamaño de la sustitución)**.** $|t\{x := s\}| = |t| + \#_x(t) \cdot |s| - \#_x(t)$

Demostración. Por inducción en t.

1. Variable, $t = y$: consideramos dos casos, dependiendo de si $y = x$ o $y \neq x$:

   1.1 $y = x$, entonces $x\{x := s\} = s$ y $\#_x(x) = 1$. Luego,

$$\begin{array}{lcl} |x\{x := s\}| & = & |s| \\ & = & 1 + |s| - 1 \\ & = & |t| + \#_x(t) \cdot |s| - \#_x(t) \end{array}$$

1.2 $y \neq x$, entonces $y\{x := s\} = y$ y $\#_x(y) = 0$. Luego,

$$\begin{array}{rcl} |y\{x := s\}| & = & |y| \\ & = & 1 \\ & = & |t| + 0 \cdot |s| - 0 \end{array}$$

2. Abstracción, $t = \lambda y.\, t'$: asumimos por α-equivalencia $y \neq x$. Luego,

$$\begin{array}{rcll} |(\lambda y.\, t')\{x := s\}| & = & |\lambda y.\, (t'\{x := s\})| & \\ & = & 1 + |t'\{x := s\}| & \\ & = & 1 + |t'| + \#_x(t') \cdot |s| - \#_x(t') & \text{por HI} \\ & = & |\lambda y.\, t'| + \#_x(\lambda y.\, t') \cdot |s| - \#_x(\lambda y.\, t') & \end{array}$$

3. Aplicación, $t = t_1\, t_2$: Luego,

$$\begin{array}{rcl} |(t_1\, t_2)\{x := s\}| & = & |(t_1\{x := s\})\, (t_2\{x := s\})| \\ & = & 1 + |t_1\{x := s\}| + |t_2\{x := s\}| \\ & = & 1 + |t_1| + |t_2| + \#_x(t_1) \cdot |s| - \#_x(t_1) + \#_x(t_2) \cdot |s| - \#_x(t_2) \\ & & \text{por HI} \\ & = & 1 + |t_1| + |t_2| + (\#_x(t_1) + \#_x(t_2)) \cdot |s| - \#_x(t_1\, t_2) \\ & = & |t_1\, t_2| + \#_x(t_1\, t_2) \cdot |s| - \#_x(t_1\, t_2) \end{array}$$

4. Demora, $t = \bullet\, t'$: Luego,

$$\begin{array}{rcll} |\bullet\, t'\{x := s\}| & = & 1 + |t'\{x := s\}| & \\ & = & 1 + |t'| + \#_x(t') \cdot |s| - \#_x(t') & \text{por HI} \\ & = & |\bullet\, t'| + \#_x(\bullet\, t') \cdot |s| - \#_x(\bullet\, t') & \end{array}$$

5. Espera, $t = t_1[y/t_2]$: asumimos por α-equivalencia $y \neq x$.

$$\begin{array}{rll} & |(t_1[y/t_2])\{x := s\}| & \\ = & |(t_1\{x := s\})[y/(t_2\{x := s\})]| & \\ = & 1 + |t_1\{x := s\}| + |t_2\{x := s\}| & \\ = & 1 + |t_1| + |t_2| + \#_x(t_1) \cdot |s| - \#_x(t_1) + \#_x(t_2) \cdot |s| - \#_x(t_2) & \text{por HI} \\ = & 1 + |t_1| + |t_2| + (\#_x(t_1) + \#_x(t_2)) \cdot |s| - \#_x(t_1\, t_2) & \\ = & |t_1[y/t_2]| + \#_x(t_1[y/t_2]) \cdot |s| - \#_x(t_1[y/t_2]) & \end{array}$$

□

Lema 4.2.7 (Cota cuadrática para un paso de reducción)**.** Si $t \to t'$ entonces $|t'| \leq |t|^2$.

Demostración. Por inducción en t.

1. Variable, $t = x$: imposible pues x no reduce.

2. Abstracción, $t = \lambda x.\, s \to \lambda x.\, s' = t'$ con $s \to s'$: por HI $|s'| \leq |s|^2$. Luego,

$$\begin{array}{rcll} |t'| & = & |\lambda x.\, s'| & \\ & = & 1 + |s'| & \\ & \leq & 1 + |s|^2 & \text{por HI} \\ & \leq & (1 + |s|)^2 & \\ & = & |t|^2 & \end{array}$$

3. Aplicación, $t = s\,u$: hay tres subcasos, dependiendo de si la reducción es interna a s, si es interna a u o en la raíz.

   3.1 Interna a s: $t = s\,u \to s'\,u$ con $s \to s'$. Por HI $|s'| \leq |s|^2$. Luego,

$$\begin{array}{rcll} |t'| & = & |s'\,u| & \\ & = & 1 + |s'| + |u| & \\ & \leq & 1 + |s|^2 + |u| & \text{por HI} \\ & \leq & (1 + |s| + |u|)^2 & \\ & = & |t|^2 & \end{array}$$

   3.2 Interna a u: $t = s\,u \to s\,u'$ con $u \to u'$. Por HI $|u'| \leq |u|^2$. Luego,

$$\begin{array}{rcll} |t'| & = & |s\,u'| & \\ & = & 1 + |s| + |u'| & \\ & \leq & 1 + |s| + |u|^2 & \text{por HI} \\ & \leq & (1 + |s| + |u|)^2 & \\ & = & |t|^2 & \end{array}$$

   3.3 En la raíz: $t = (\lambda x.\, s)\,u \to \bullet\, s\{x := u\} = t'$. Luego,

$$\begin{array}{rcll} |t'| & = & |\bullet\, s\{x := u\}| & \\ & = & 1 + |s\{x := u\}| & \\ & = & 1 + |s| + \#_x(s) \cdot |u| - \#_x(s) & \text{por Lema 4.2.6} \\ & \leq & 1 + |s| + |s||u| & \text{pues } \#_x(s) \leq |s| \\ & & & \text{por Observación 4.2.5} \\ & \leq & (2 + |s| + |u|)^2 & \\ & = & |(\lambda x.\, s)\,u|^2 & \\ & = & |t|^2 & \end{array}$$

4. Demora, $t = \bullet\, s \to \bullet\, s' = t'$ con $s \to s'$: por HI $|s'| \leq |s|^2$. Luego,

$$\begin{array}{rcll} |t'| & = & |\bullet\, s'| & \\ & = & 1 + |s'| & \\ & \leq & 1 + |s|^2 & \text{por HI} \\ & \leq & (1 + |s|)^2 & \\ & = & |t|^2 & \end{array}$$

5. Espera, $t = s[x/u]$: hay tres subcasos, dependiendo de si la reducción es interna a s, si es interna a u o en la raíz.

   5.1 Interna a s: $t = s[x/u] \to s'[x/u]$ con $s \to s'$. Por HI $|s'| \leq |s|^2$. Luego,

$$\begin{array}{rcll} |t'| & = & |s'[x/u]| & \\ & = & 1 + |s'| + |u| & \\ & \leq & 1 + |s|^2 + |u| & \text{por HI} \\ & \leq & (1 + |s| + |u|)^2 & \\ & = & |t|^2 & \end{array}$$

5.2 Interna a u: $t = s[x/u] \to s[x/u']$ con $u \to u'$. Por HI $|u'| \leq |u|^2$. Luego,

$$\begin{array}{lcll}
|t'| & = & |s[x/u']| & \\
 & = & 1 + |s| + |u'| & \\
 & \leq & 1 + |s| + |u|^2 & \text{por HI} \\
 & \leq & (1 + |s| + |u|)^2 & \\
 & = & |t|^2 &
\end{array}$$

5.3 En la raíz: $t = s[x/\bullet\, u] \to \bullet\, s\{x := u\} = t'$. Luego,

$$\begin{array}{lcll}
|t'| & = & |\bullet\, s\{x := u\}| & \\
 & = & 1 + |s\{x := u\}| & \\
 & = & 1 + |s| + \#_x(s) \cdot |u| - \#_x(s) & \text{por Lema 4.2.6} \\
 & \leq & 1 + |s| + |s||u| & \text{pues } \#_x(s) \leq |s| \\
 & & & \text{por Observación 4.2.5} \\
 & \leq & (2 + |s| + |u|)^2 & \\
 & = & |s[x/\bullet\, u]|^2 & \\
 & = & |t|^2 &
\end{array}$$

□

Lema 4.2.8 (Cota doble exponencial para n pasos de reducción)**.** Si $t \to^n t'$ entonces $|t'| \leq |t|^{2^n}$.

Demostración. Por inducción en n.

1. $n = 0$: luego $t = t'$. Por lo tanto, $|t'| = |t| \leq |t|^1 = |t|^{2^0}$.

2. Suponemos que vale para n y veamos que vale con $n + 1$. Luego, $t \to s \to^n t'$.

$$\begin{array}{lcll}
|t'| & \leq & |s|^{2^n} & \text{por HI} \\
 & \leq & (|t|^2)^{2^n} & \text{por Lema 4.2.7} \\
 & = & |t|^{2 \cdot 2^n} & \\
 & = & |t|^{2^{n+1}} &
\end{array}$$

□

Corolario 4.2.9. Si $t \to^{\leq n} t'$ entonces $|t'| \leq |t|^{2^n}$.

Lema 4.2.10 (Productividad en un paso)**.** Si $\Gamma \rhd t : \bullet\, A$ entonces se da alguna de las dos condiciones siguientes:

1. t está en forma normal y está listo,

2. existe t' tal que $t \to^{\leq |t|} \bullet\, t'$.

Demostración. Por inducción en t.

1. $t = x$: está en forma normal y está listo.

2. $t = \lambda x.\, s$: imposible pues el tipo de una abstracción nunca puede ser de la forma $\bullet\, A$.

3. $t = s\,u$: luego,
$$\frac{\Gamma \rhd s : A \to B \quad \Gamma \rhd u : A}{\Gamma \rhd s\,u : \bullet\, B}\,\mathrm{T}^{\bullet}\textsc{App}$$
Entonces, por Observación 4.2.4, $s = \lambda x.\, s'$. Luego, $t = (\lambda x.\, s')\, u \to^{1} \bullet\, s'\{x := u\}$ y $1 \leq 1 + |s| + |u| = |s\,u| = |t|$.

4. $t = \bullet\, s$: inmediato pues $t = \bullet\, s \to^{\leq 0} \bullet\, s$ y $0 \leq |t|$.

5. $t = s[x/u]$: luego,
$$\frac{\Gamma, x : A \rhd s : B \quad \Gamma \rhd u : \bullet\, A}{\Gamma \rhd s[x/u] : \bullet\, B}\,\mathrm{T}^{\bullet}\textsc{Wait}$$
Por HI $u \to^{\leq |u|} \bullet\, u'$. Por lo tanto, $t = s[x/u] \to^{\leq |u|} s[x/\bullet\, u'] \to^{1} \bullet\, s\{x := u'\}$. Observemos que $|u| + 1 \leq 1 + |s| + |u| = |s[x/u]| = |t|$.

□

Definición 4.2.11 (Potencia de "$\bullet$")**.** Definimos la n-ésima *potencia* de $\bullet$ de la siguiente forma.

1. Para los tipos:
$$\begin{array}{lcl} \bullet^{0}\, A & \stackrel{\text{def}}{=} & A \\ \bullet^{n+1}\, A & \stackrel{\text{def}}{=} & \bullet\,(\bullet^{n}\, A) \end{array}$$

2. Para los términos:
$$\begin{array}{lcl} \bullet^{0}\, t & \stackrel{\text{def}}{=} & t \\ \bullet^{n+1}\, t & \stackrel{\text{def}}{=} & \bullet\,(\bullet^{n}\, t) \end{array}$$

Definición 4.2.12 (Grado de A)**.** Definimos el *grado* de un tipo A como la cantidad de "$\bullet$" que aparecen en A. Formalmente:
$$\begin{array}{lcl} d(\alpha) & \stackrel{\text{def}}{=} & 0 \\ d(A \to B) & \stackrel{\text{def}}{=} & d(A) + d(B) \\ d(\bullet\, A) & \stackrel{\text{def}}{=} & 1 + d(A) \end{array}$$

Lema 4.2.13 (Si el grado es cero, el término está en forma normal)**.** Si $\Gamma \rhd t : A$ y $d(A) = 0$, entonces t está en forma normal.

Demostración. Observemos que A no puede ser de la forma $\bullet\, A'$ pues sabemos que $d(A) = 0$. Consideramos dos casos, dependiendo de la forma de A:

1. Si $A = \alpha$: observemos que t debe ser una variable pues sólo la regla $\mathrm{T}^{\bullet}\textsc{Var}$ permite concluir que un término tiene un tipo de la forma α. Luego $t = x$ está en forma normal.

2. Si $A = (B \to C)$: observemos que t debe ser una abstracción, pues sólo la regla $\mathrm{T}^{\bullet}\textsc{Abs}$ permite concluir que un término tiene un tipo de la forma $B \to C$. Luego $t = \lambda x.\, s$ y tenemos:
$$\frac{\Gamma, x : B \rhd s : C}{\Gamma \rhd \lambda x.\, s : B \to C}\,\mathrm{T}^{\bullet}\textsc{Abs}$$
Por HI, s está en forma normal. Por lo tanto, $t = \lambda x.\, s$ también está en forma normal.

□

Definición 4.2.14 (Función de crecimiento)**.** Definimos la función $\mathbf{G} : \mathbb{N}_0 \to \mathbb{N}_0$ por recursión en el segundo argumento:

$$\begin{array}{lcl} \mathbf{G}(m,0) & = & 0 \\ \mathbf{G}(m,n+1) & = & m + \mathbf{G}(m^{2^m}, n) \end{array}$$

Lema 4.2.15 (Monotonía de la función de crecimiento)**.** Si $m \leq m'$ entonces $\mathbf{G}(m,n) \leq \mathbf{G}(m',n)$.

Demostración. Por inducción en n.

1. $n = 0$: entonces $\mathbf{G}(m,0) = \mathbf{G}(m',0) = 0$ y en efecto $0 \leq 0$.

2. Suponemos que vale para n y veamos que vale para $n+1$. En efecto

$$\begin{array}{lcll} \mathbf{G}(m,n+1) & = & m + \mathbf{G}(m^{2^m}, n) & \\ & \leq & m' + \mathbf{G}(m^{2^m}, n) & \text{pues } m \leq m' \\ & \leq & m' + \mathbf{G}(m'^{2^{m'}}, n) & \text{por HI pues } m^{2^m} \leq m'^{2^{m'}} \text{ pues } m \leq m' \\ & = & m' + \mathbf{G}(m'^{2^{m'}}, n) & \\ & = & \mathbf{G}(m', n+1) & \end{array}$$

□

Teorema 4.2.16 (Terminación cronometrada)**.** Si $\Gamma \rhd t : \bullet^n A$ y $d(A) = 0$, entonces existen un entero $k \in \mathbb{N}_0$ y un término $t' \in \Lambda_\bullet^\to$ tales que:

1. $0 \leq k \leq n$

2. $\bullet^k t'$ está en forma normal

3. $t \to^{\leq \mathbf{G}(|t|,k)} \bullet^k t'$

Demostración. Por inducción en n.

1. $n = 0$: sabemos que $\Gamma \rhd t : A$ con $d(A) = 0$. Por Lema 4.2.13, t está en forma normal. Tomemos $k := 0$ y $t' := t$. Luego, $t \to^0 t = t'$ y $0 \leq 0 = \mathbf{G}(|t|, 0)$.

2. Suponemos que vale para n y veamos que vale para $n+1$. Sabemos que $\Gamma \rhd t : \bullet^{n+1} A$, es decir, $\Gamma \rhd t : \bullet (\bullet^n A)$. Por el Lema 4.2.10 hay dos casos:

   2.1 Si t está en forma normal y está listo. Tomando $k := 0$ y $t' := t$ tenemos $t \to^0 t = t'$ y $0 \leq 0 = \mathbf{G}(|t|, 0)$.

   2.2 Si existe t'' tal que $t \to^{\leq |t|} \bullet t''$. Por Lema 4.2.8, $|\bullet t''| \leq |t|^{2^{|t|}}$. En particular, $|t''| \leq |t|^{2^{|t|}}$. Como $\Gamma \rhd t : \bullet^{n+1} A$, por preservación de tipos (Proposición 3.3.3) $\Gamma \rhd \bullet t'' : \bullet^{n+1} A$. Como este juicio sólo se puede haber derivado usando la regla T$^\bullet$Delay, tenemos que $\Gamma \rhd t'' : \bullet^n A$. Por HI existen $k' \in \mathbb{N}_0$ y $t' \in \Lambda_\bullet^\to$ tales que $0 \leq k' \leq n$, donde $\bullet^{k'} t'$ está en forma normal y además:

$$t'' \to^{\leq \mathbf{G}(|t''|,k')} \bullet^{k'} t'$$

Luego,

$$t \to^{\leq |t|}_{\bullet} t'' \to^{\leq \mathbf{G}(|t''|,k')}_{\bullet^{k'+1}} t'$$

Tomando $k := k' + 1$ sabemos:

$$\begin{array}{rcll} |t| + \mathbf{G}(|t''|, k') & \leq & |t| + \mathbf{G}(|t|^{2^{|t|}}, k') & \text{pues } |t''| \leq |t|^{2^{|t|}} \text{ y por Lema 4.2.15} \\ & = & \mathbf{G}(|t|, k' + 1) & \text{por definición de } \mathbf{G} \\ & = & \mathbf{G}(|t|, k) & \end{array}$$

Por lo tanto $0 \leq k = k' + 1 \leq n + 1$ y además:

$$t \to^{\leq \mathbf{G}(|t|,k)}_{\bullet^k} t'$$

$\square$

5. TEMPORIZACIÓN

El cálculo-$\lambda^\bullet$ estudiado en los capítulos anteriores es una variante del cálculo-λ con anotaciones temporales. Una pregunta natural es si, dado un término tipable del cálculo-λ, es posible "temporizar" el término para que sea tipable en el cálculo-$\lambda^\bullet$. Por ejemplo, el término $f\,x\,y$ es tipable en el cálculo-λ. Más precisamente, vale el siguiente juicio de tipado:

$$f : A \to B \to C,\, x : A,\, y : B \vdash f\,x\,y : C$$

Observemos que dicho término no es tipable en el cálculo-$\lambda^\bullet$ debido a que $f\,x$ no tiene tipo función, sino que su tipo es $\bullet\,(B \to C)$. Sin embargo, el término $f\,x\,y$ se puede "temporizar" escribiéndolo como $(g\,y)[g/f\,x]$, que (intuitivamente) describe el mismo cómputo, pero es tipable en el cálculo-$\lambda^\bullet$. Más precisamente, vale el siguiente juicio de tipado:

$$f : A \to B \to C,\, x : A,\, y : B \rhd (g\,y)[g/f\,x] : \bullet\bullet C$$

En este capítulo definimos la noción de término *temporizable* (Definición 5.0.6) y estudiamos la pregunta de si todos los términos tipables en cálculo-λ son temporizables. Conjeturamos que la respuesta a esta pregunta es negativa, es decir, que existen λ-términos tipables que no son temporizables. Sin embargo, no conocemos una demostración de este teorema.

Como manera de abordar esta pregunta, definimos una noción más restrictiva de término *fuertemente temporizable* (Definición 5.0.6). El resultado principal de este capítulo es que existen términos que **no** son fuertemente temporizables (Teorema 5.0.21).

Para demostrar dicho resultado, definimos un sistema auxiliar, al que denominamos cálculo-$\lambda^?_\bullet$ que predica sobre términos del cálculo-λ y les asigna tipos del cálculo-$\lambda^\bullet$ en los que las demoras ($\bullet$) vienen decoradas con *incógnitas* que representan el número de veces que debe repetirse la demora. Por ejemplo, $\bullet^{?a} A$ representa el tipo A acompañado de un número de demoras estipulado por el valor de la incógnita $?a$, de tal modo que si la incógnita toma el valor $?a = 3$ se tiene que $\bullet^{?a} A = \bullet\bullet\bullet A$.

En el cálculo-$\lambda^?_\bullet$, los juicios están acompañados de sistemas de ecuaciones entre las incógnitas. Definimos primero este sistema de tipos (Definición 5.0.9) y demostramos que es correcto (Proposición 5.0.16) y completo (Proposición 5.0.19), en el sentido de que un λ-término es fuertemente temporizable si y sólo si es tipable en el cálculo-$\lambda^?_\bullet$ con un sistema de ecuaciones soluble. El teorema principal ya mencionado (Teorema 5.0.21) se obtiene como consecuencia de estos resultados.

Definición 5.0.1 (Función de olvido)**.** Si $t \in \Lambda_\bullet$, definimos $t^\circ \in \Lambda$ por inducción en t de la siguiente forma:

$$\begin{array}{lcl} x^\circ & = & x \\ (\lambda x.\, t)^\circ & = & \lambda x.\, t^\circ \\ (t\,s)^\circ & = & t^\circ\, s^\circ \\ (\bullet\, t)^\circ & = & t^\circ \\ t[x/s]^\circ & = & t^\circ\{x := s^\circ\} \end{array}$$

Definición 5.0.2 (Aplicación múltiple)**.** Dados términos t, s y un número natural $n \in \mathbb{N}_0$, se define $t\,@^n\,s$ por recursión en n de la siguiente manera:

$$\begin{array}{lcll} t\,@^0\,s & \stackrel{\text{def}}{=} & t\,s & \\ t\,@^{n+1}\,s & \stackrel{\text{def}}{=} & (x\,@^n\,s)[x/t] & \text{donde } x \notin \mathsf{fv}(s) \end{array}$$

Observación 5.0.3. $(t\,@^n\,s)^\circ = t^\circ\,s^\circ$

Definición 5.0.4 (Términos regulares)**.** Un término $t \in \Lambda_\bullet$ es regular si vale $\texttt{regular}(t)$ de acuerdo con las siguientes reglas:

$$\frac{}{\texttt{regular}(x)} \qquad \frac{\texttt{regular}(t)}{\texttt{regular}(\lambda x.\,t)} \qquad \frac{\texttt{regular}(t) \quad \texttt{regular}(s)}{\texttt{regular}(t\,@^n\,s)} \qquad \frac{\texttt{regular}(t)}{\texttt{regular}(\bullet\,t)}$$

Por ejemplo, $(x\,y)[x/z]$ es regular pues se puede escribir como $z\,@^1\,y$, en tanto que los términos $(y\,x)[x/z]$ y $(x\,x)[x/z]$ no son regulares.

Observación 5.0.5. Si t es un término regular, se da exactamente uno de los tres siguientes casos:

1. $t = \bullet^n\,x$.

2. $t = \bullet^n\,\lambda x.\,s$ con s regular.

3. $t = \bullet^n\,(s\,@^k\,u)$ con s y u regulares.

Definición 5.0.6 (Términos temporizables)**.**

1. Un término $t \in \Lambda^{\to}$ tipable del cálculo-λ es *temporizable* si existe un término $t' \in \Lambda_\bullet^{\to}$ tipable del cálculo-$\lambda^\bullet$ tal que $(t')^\circ = t$.

2. Un término $t \in \Lambda^{\to}$ tipable del cálculo-λ es *fuertemente temporizable* si existe un $t' \in \Lambda_\bullet^{\to}$ tipable del cálculo-$\lambda^\bullet$ tal que $(t')^\circ = t$, donde además t' es un término regular.

Definición 5.0.7 (Cálculo-$\lambda_\bullet^?$)**.** Definimos un nuevo cálculo-$\lambda_\bullet^?$ con los términos del cálculo-λ, es decir:

$$t ::= x \mid \lambda x.\,t \mid t\,t$$

Suponemos dado un conjunto infinito de tipos base $\{\alpha, \beta, \ldots\}$. Los conjuntos de *tipos* y *tipos demorados* se definen de manera mutuamente recursiva por la siguiente gramática:

$$\begin{array}{lcll} \text{Tipos} & A & ::= & \alpha \mid \mathbb{A} \to \mathbb{A} \\ \text{Tipos demorados} & \mathbb{A} & ::= & \bullet^{?a}\,A \end{array}$$

Generalmente llamamos *tipos* tanto a los tipos (a secas) como a los tipos demorados. Un *contexto de tipado* es un conjunto de asignaciones de variables a tipos demorados, de la forma:

$$\Gamma = (x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n)$$

Suponemos dado un conjunto infinito numerable de *incógnitas* $\mathcal{I} = \{?a, ?b, ?c, \ldots\}$ el conjunto de todas las incógnitas. Una *expresión aritmética* es una expresión e dada por la siguiente gramática:

$$\begin{array}{lcll} e & ::= & n & \text{con } n \in \mathbb{N}_0 \\ & \mid & ?a & \text{con } ?a \in \mathcal{I} \\ & \mid & e + e & \end{array}$$

Una *ecuación* es un par $(e_1 \stackrel{?}{=} e_2)$ de expresiones aritméticas. Notamos $\mathcal{E}$ al conjunto de todas las ecuaciones.

Definición 5.0.8 (Función de compatibilidad)**.** Definimos una función parcial **Eq** que dado un tipo (o un tipo demorado) devuelve un conjunto de ecuaciones:

$$\begin{array}{lcll}
\mathbf{Eq}(\alpha,\alpha) & = & \varnothing & \\
\mathbf{Eq}(\mathbb{A}\to\mathbb{B},\mathbb{A}'\to\mathbb{B}') & = & \mathbf{Eq}(\mathbb{A},\mathbb{A}')\cup\mathbf{Eq}(\mathbb{B},\mathbb{B}') & \\
\mathbf{Eq}(\alpha,\beta) & = & \text{(indefinido)} & \text{si } \alpha\neq\beta \\
\mathbf{Eq}(\alpha,\mathbb{A}\to\mathbb{B}) & = & \text{(indefinido)} & \\
\mathbf{Eq}(\mathbb{A}\to\mathbb{B},\alpha) & = & \text{(indefinido)} & \\
\mathbf{Eq}(\bullet^{?a}\,\mathbb{A},\bullet^{?b}\,\mathbb{B}) & = & \{?a \stackrel{?}{=} ?b\}\cup\mathbf{Eq}(\mathbb{A},\mathbb{B}) &
\end{array}$$

Definición 5.0.9 (Sistema de tipos $\lambda^?_\bullet$)**.** Los *juicios de tipado* son de la forma $\Gamma \blacktriangleright_E t : A$ donde $E \subseteq \mathcal{E}$ es un conjunto finito de ecuaciones. Los juicios válidos se obtienen inductivamente a través de las siguientes reglas de tipado:

$$\frac{E = \mathbf{Eq}(A_1, A_2)}{\Gamma, x : \bullet^{?a}\, A_1 \blacktriangleright_{E\cup\{?b \stackrel{?}{=} ?a+?c\}} x : \bullet^{?b}\, A_2}\ \mathrm{T}^?_\bullet\textsc{Var} \qquad \frac{\Gamma, x : \mathbb{A} \blacktriangleright_E t : \mathbb{B}}{\Gamma \blacktriangleright_E \lambda x.\, t : \bullet^{?a}\,(\mathbb{A}\to\mathbb{B})}\ \mathrm{T}^?_\bullet\textsc{Abs}$$

$$\frac{\Gamma \blacktriangleright_{E_1} t : \bullet^{?a}\,(\mathbb{A}_1 \to \bullet^{?b}\,\mathbb{B}) \qquad \Gamma \blacktriangleright_{E_2} s : \mathbb{A}_2 \qquad E_3 = \mathbf{Eq}(\mathbb{A}_1,\mathbb{A}_2)}{\Gamma \blacktriangleright_{E_1\cup E_2\cup E_3\cup\{?c \stackrel{?}{=} 1+?a+?b+?d\}} t\, s : \bullet^{?c}\,\mathbb{B}}\ \mathrm{T}^?_\bullet\textsc{App}$$

Observación 5.0.10. Si $\mathbf{Eq}(\mathbb{A}_1,\mathbb{A}_2)$ no está definido, no es posible aplicar las reglas $\mathrm{T}^?_\bullet\textsc{Var}$ y $\mathrm{T}^?_\bullet\textsc{App}$. Por ejemplo, el juicio $x : \bullet^{?a}\,\alpha \blacktriangleright_E x : \bullet^{?b}\,\alpha$ es válido con $E = \{?a \stackrel{?}{=} ?b\}$, pero no hay un juicio válido de la forma $x : \bullet^{?a}\,\alpha \blacktriangleright_E x : \bullet^{?b}\,\beta$ si $\alpha \neq \beta$.

Ejemplo 5.0.11. Sea $\mathtt{I}$ la identidad, definida como $\mathtt{I} \stackrel{\text{def}}{=} \lambda x.\, x$, y las siguientes ecuaciones:

$$\begin{array}{lcl}
E_1 & := & \{?g \stackrel{?}{=} ?j, ?h \stackrel{?}{=} ?k, ?i \stackrel{?}{=} ?f+?o\} \\
E_2 & := & E_1 \\
E_3 & := & \{?d \stackrel{?}{=} ?c+?p\} \\
E_4 & := & E_3 \\
E_5 & := & E_2 \cup E_4 \cup \{?f \stackrel{?}{=} ?e, ?g \stackrel{?}{=} ?c, ?h \stackrel{?}{=} ?d\} \cup \{?m \stackrel{?}{=} 1+?l+?i+?q\} \\
E_6 & := & \{?b \stackrel{?}{=} ?a+?r\} \\
E_7 & := & E_5 \cup E_6 \cup \{?j \stackrel{?}{=} ?b\} \cup \{?n \stackrel{?}{=} 1+?m+?k+?s\}
\end{array}$$

Luego tenemos:

$$\pi_1 = \left(\frac{\dfrac{}{z : \bullet^{?a}\,\alpha, y : \bullet^{?f}\,(\bullet^{?g}\,\alpha \to \bullet^{?h}\,\alpha) \blacktriangleright_{E_1} y : \bullet^{?i}\,(\bullet^{?j}\,\alpha \to \bullet^{?k}\,\alpha)}\ \mathrm{T}^?_\bullet\textsc{Var}}{z : \bullet^{?a}\,\alpha \blacktriangleright_{E_2} \mathtt{I} : \bullet^{?l}\,(\bullet^{?f}\,(\bullet^{?g}\,\alpha \to \bullet^{?h}\,\alpha) \to \bullet^{?i}\,(\bullet^{?j}\,\alpha \to \bullet^{?k}\,\alpha))}\ \mathrm{T}^?_\bullet\textsc{Abs} \right)$$

$$\pi_2 = \left(\frac{\dfrac{}{z : \bullet^{?a}\,\alpha, x : \bullet^{?c}\,\alpha \blacktriangleright_{E_3} x : \bullet^{?d}\,\alpha}\ \mathrm{T}^?_\bullet\textsc{Var}}{z : \bullet^{?a}\,\alpha \blacktriangleright_{E_4} \mathtt{I} : \bullet^{?e}\,(\bullet^{?c}\,\alpha \to \bullet^{?d}\,\alpha)}\ \mathrm{T}^?_\bullet\textsc{Abs} \right)$$

$$
\dfrac{\dfrac{\begin{matrix}\vdots & \vdots \\ \pi_1 & \pi_2\end{matrix}}{z : \bullet^{?a}\,\alpha \blacktriangleright_{E_5} \mathtt{I\,I} : \bullet^{?m}\,(\bullet^{?j}\,\alpha \to \bullet^{?k}\,\alpha)}\;\mathrm{T}^{?}_{\bullet}\textsc{App} \qquad \dfrac{}{z : \bullet^{?a}\,\alpha \blacktriangleright_{E_6} z : \bullet^{?b}\,\alpha}\;\mathrm{T}^{?}_{\bullet}\textsc{Var}}{z : \bullet^{?a}\,\alpha \blacktriangleright_{E_7} \mathtt{I\,I}\, z : \bullet^{?n}\,\alpha}\;\mathrm{T}^{?}_{\bullet}\textsc{App}
$$

Definición 5.0.12 (Valuaciones)**.** Una *valuación* es una función $V : \mathcal{I} \to \mathbb{N}_0$ que a cada incógnita le asocia un número natural. Las valuaciones se extienden para aplicarse a expresiones aritméticas de la manera esperable:

$$
\begin{array}{lcl}
V(n) & \stackrel{\text{def}}{=} & n \\
V(e_1 + e_2) & \stackrel{\text{def}}{=} & V(e_1) + V(e_2)
\end{array}
$$

Las valuaciones se extienden también para aplicarse a tipos y tipos demorados:

$$
\begin{array}{lcl}
V(\alpha) & = & \alpha \\
V(\mathbb{A} \to \mathbb{B}) & = & V(\mathbb{A}) \to V(\mathbb{B}) \\
V(\bullet^{?a}\,A) & = & \bullet^{V(?a)}\,V(A)
\end{array}
$$

Por último, las valuaciones se extienden para aplicarse a contextos de tipado de la siguiente manera. Si $\Gamma = (x_1 : \mathbb{A}, \ldots, x_n : \mathbb{A}_n)$ entonces $V(\Gamma) = (x_1 : V(\mathbb{A}), \ldots, x_n : V(\mathbb{A}_n))$.

Definición 5.0.13 (Solubilidad)**.** Una valuación V es *solución* de un conjunto de ecuaciones $E \subseteq \mathcal{E}$ si y sólo si para toda ecuación $(e_1 \stackrel{?}{=} e_2) \in E$ se tiene que $V(e_1) = V(e_2)$. Notamos $V \vDash E$ si V es solución de E. Un conjunto de ecuaciones E es *soluble* si y sólo si existe una valuación V tal que $V \vDash E$.

Observación 5.0.14.

1. Si $\mathbf{Eq}(\mathbb{A}, \mathbb{B})$ está definido y $V \vDash \mathbf{Eq}(\mathbb{A}, \mathbb{B})$, entonces $V(\mathbb{A}) = V(\mathbb{B})$.

2. $V \vDash E_1 \cup E_2$ si y sólo si $V \vDash E_1$ y $V \vDash E_2$

Lema 5.0.15 (Tipo de la aplicación múltiple)**.** Son equivalentes:

1. $\Gamma \rhd t \mathbin{@^n} s : C$

2. Existen tipos A, B tales que:

$$
C = \bullet^{n+1}\,B \qquad \Gamma \rhd t : \bullet^{n}\,(A \to B) \qquad \Gamma \rhd s : A
$$

Demostración.

- ($1 \implies 2$) Por inducción en n:

  1. $n = 0$: sabemos $\Gamma \rhd t \mathbin{@^0} s : C$, es decir, $\Gamma \rhd t\,s : C$. Luego,

$$
\dfrac{\Gamma \rhd t : A \to B \qquad \Gamma \rhd s : A}{\Gamma \rhd t\,s : \bullet B}\;\mathrm{T}^{\bullet}\textsc{App}
$$

  Por lo que se cumplen las tres condiciones:

$$
C = \bullet\, B = \bullet^1\,B \qquad \Gamma \rhd t : \bullet^0\,(A \to B) \qquad \Gamma \rhd s : A
$$

2. Suponemos que vale para n y veamos que vale para $n+1$. Sabemos $\Gamma \rhd t \mathbin{@^{n+1}} s : C$, es decir, $\Gamma \rhd (x \mathbin{@^n} s)[x/t] : C$. Tenemos entonces:

$$\dfrac{\Gamma, x : A \rhd x \mathbin{@^n} s : B \qquad \Gamma \rhd t : \bullet\, A}{\Gamma \rhd (x \mathbin{@^n} s)[x/t] : \bullet\, B}\ \mathrm{T}^{\bullet}\textsc{Wait}$$

Como vale $\Gamma, x : A \rhd x \mathbin{@^n} s : B$, por HI existen D, E tales que:

$$B = \bullet^{n+1} D \qquad \Gamma, x : A \rhd x : \bullet^{n} (E \to D) \qquad \Gamma \rhd s : E$$

Como el juicio $\Gamma, x : A \rhd x : \bullet^n (E \to D)$ sólo se puede derivar usando la regla $\mathrm{T}^{\bullet}\textsc{Var}$, debe ser $A = \bullet^n (E \to D)$. Tenemos entonces:

2.1 $C = \bullet\, B = \bullet\, \bullet^{n+1} D = \bullet^{n+2} D$

2.2 $\Gamma \rhd t : \bullet\, A$ por lo cual $\Gamma \rhd t : \bullet\,\bullet^n (E \to D)$, es decir $\Gamma \rhd t : \bullet^{n+1} (E \to D)$

2.3 $\Gamma \rhd s : E$

- ($2 \implies 1$) Por inducción en n:

1. $n = 0$: sabemos $\Gamma \rhd t : \bullet^0 (A \to B)$, es decir, $\Gamma \rhd t : (A \to B)$, y $\Gamma \rhd s : A$. Luego,

$$\dfrac{\Gamma \rhd t : A \to B \qquad \Gamma \rhd s : A}{\Gamma \rhd t\, s : \bullet\, B}\ \mathrm{T}^{\bullet}\textsc{App}$$

Notar que $\Gamma \rhd t\, s : \bullet\, B$ es lo mismo que $\Gamma \rhd t \mathbin{@^0} s : \bullet^1 B$.

2. Suponemos que vale para n y veamos que vale para $n + 1$. Sabemos $\Gamma \rhd t : \bullet^{n+1} (A \to B)$, es decir, $\Gamma \rhd t : \bullet\,\bullet^n (A \to B)$, y $\Gamma \rhd s : A$. Sabemos además por $\mathrm{T}^{\bullet}\textsc{Var}$,

$$\Gamma, x : \bullet^n (A \to B) \rhd x : \bullet^n (A \to B)$$

Notemos que $\Gamma, x : \bullet^n (A \to B) \rhd s : A$ por Lema 3.3.1. Por HI tenemos

$$\Gamma, x : \bullet^n (A \to B) \rhd x \mathbin{@^n} s : \bullet^{n+1} B$$

Por lo tanto, por $\mathrm{T}^{\bullet}\textsc{Wait}$, como vale $\Gamma \rhd t : \bullet\,\bullet^n A \to B$, vale

$$\Gamma \rhd (x \mathbin{@^{n+1}} s)[x/t] : \bullet^{n+2} B$$

$\square$

Proposición 5.0.16 (Correctitud del cálculo-$\lambda^?_\bullet$)**.** Si $\Gamma \blacktriangleright_E t : \mathbb{A}$ y $V \vDash E$, entonces existe un término regular $t' \in \Lambda_\bullet$ tal que $V(\Gamma) \rhd t' : V(\mathbb{A})$ donde $(t')^\circ = t$.

Demostración. Procedemos por inducción en la derivación del juicio $\Gamma \blacktriangleright_E t : \mathbb{A}$:

1. $\mathrm{T}^?_\bullet\textsc{Var}$: sabemos que vale $\Gamma, x : \bullet^{?a} A_1 \blacktriangleright_{E \cup \{?b \overset{?}{=} ?a + ?c\}} x : \bullet^{?b} A_2$ donde $E = \mathbf{Eq}(A_1, A_2)$. Sea $V \vDash \mathbf{Eq}(A_1, A_2) \cup \{?b \overset{?}{=} ?a + ?c\}$, en particular, $V \vDash \mathbf{Eq}(A_1, A_2)$ y $V \vDash \{?b \overset{?}{=} ?a + ?c\}$ por Observación 5.0.14.

Luego tenemos $V(A_1) = V(A_2)$ por Observación 5.0.14. Por lo tanto vale:

$$\cfrac{\cfrac{\cfrac{}{V(\Gamma), x : \bullet^{?a} V(A_1) \rhd x : \bullet^{?a} V(A_2)}\ \mathrm{T}^{\bullet}\textsc{Var}}{\vdots \Big\} V(?c) \text{ veces}}\ \mathrm{T}^{\bullet}\textsc{Delay}}{V(\Gamma), x : \bullet^{V(?a)} V(A_1) \rhd \bullet^{V(?c)} x : \bullet^{V(?a)+V(?c)} V(A_2)}\ \mathrm{T}^{\bullet}\textsc{Delay}$$

Tomando $t' := \bullet^{V(?c)} x$ se tiene que t' es regular, porque resulta de aplicar demoras a una variable. Para concluir, basta observar que $V(?a) + V(?c) = V(?b)$ pues $V \vDash \{?a+?c \stackrel{?}{=} ?b\}$.

2. $\mathrm{T}^?_\bullet$Abs: luego, $t = \lambda x.\, s$ y sabemos,

$$\cfrac{\Gamma, x : \mathbb{A} \blacktriangleright_E s : \mathbb{B}}{\Gamma \blacktriangleright_E \lambda x.\, s : \bullet^{?a} (\mathbb{A} \to \mathbb{B})}\ \mathrm{T}^?_\bullet\textsc{Abs} \qquad V \vDash E$$

Por HI existe un término regular $s' \in \Lambda_\bullet$ tal que $V(\Gamma), x : V(\mathbb{A}) \rhd s' : V(\mathbb{B})$ y $(s')^\circ = s$. Por lo tanto, tomando $t' := \bullet^{V(?a)} \lambda x.\, s'$ se tiene que $(t')^\circ = \lambda x.\, (s')^\circ = \lambda x.\, s = t$. Y además:

$$\cfrac{\cfrac{\cfrac{\cfrac{\text{HI}}{V(\Gamma), x : V(\mathbb{A}) \rhd s' : V(\mathbb{B})}}{V(\Gamma) \rhd \lambda x.\, s' : V(\mathbb{A}) \to V(\mathbb{B})}\ \mathrm{T}^{\bullet}\textsc{Abs}}{\vdots \Big\} V(?a) \text{ veces}}\ \mathrm{T}^{\bullet}\textsc{Delay}}{V(\Gamma) \rhd \bullet^{V(?a)} \lambda x.\, s' : \bullet^{V(?a)} V(\mathbb{A}) \to V(\mathbb{B})}\ \mathrm{T}^{\bullet}\textsc{Delay}$$

Tomando $t' := \bullet^{V(?a)} \lambda x.\, s'$, se tiene que t' es regular pues resulta de aplicar demoras a una abstracción cuyo cuerpo s' es regular. Para concluir, basta notar que $\bullet^{V(?a)} V(\mathbb{A}) \to V(\mathbb{B}) = V(\bullet^{?a} (\mathbb{A} \to \mathbb{B}))$.

3. $\mathrm{T}^?_\bullet$App: luego $t = s\, u$ y sabemos,

$$\cfrac{\Gamma \blacktriangleright_{E_1} s : \bullet^{?a} (\mathbb{A}_1 \to \bullet^{?b} \mathbb{B}) \qquad \Gamma \blacktriangleright_{E_2} u : \mathbb{A}_2 \qquad E_3 = \mathbf{Eq}(\mathbb{A}_1, \mathbb{A}_2)}{\Gamma \blacktriangleright_{E_1 \cup E_2 \cup E_3 \cup \{?c \stackrel{?}{=} 1+?a+?b+?d\}} s\, u : \bullet^{?c} \mathbb{B}}\ \mathrm{T}^?_\bullet\textsc{App}$$

Sea $V \vDash E_1 \cup E_2 \cup E_3 \cup \{?c \stackrel{?}{=} 1+?a+?b+?d\}$, en particular, $V \vDash E_1$ y $V \vDash E_2$ por Observación 5.0.14. Por HI sobre la primera premisa existe un término regular $s' \in \Lambda_\bullet$ tal que

$$V(\Gamma) \rhd s' : \bullet^{V(?a)} (V(\mathbb{A}_1) \to \bullet^{V(?b)} V(\mathbb{B})) \ \text{ y } (s')^\circ = s.$$

y por HI sobre la segunda premisa existe un término regular $u' \in \Lambda_\bullet$ tal que

$$V(\Gamma) \rhd u' : V(\mathbb{A}_2) \ \text{ y } (u')^\circ = u.$$

Como también $V \vDash E_3 = \mathbf{Eq}(\mathbb{A}_1, \mathbb{A}_2)$ tenemos $V(\mathbb{A}_1) = V(\mathbb{A}_2)$ por Observación 5.0.14. Por lo tanto tenemos:

$$\pi = \left(\cfrac{V(\Gamma) \rhd s' : \bullet^{V(?a)} (V(\mathbb{A}_1) \to \bullet^{V(?b)} V(\mathbb{B})) \qquad \Gamma \rhd u' : V(\mathbb{A}_2)}{V(\Gamma) \rhd s' @^{V(?a)} u' : \bullet \bullet^{V(?a)} \bullet^{V(?b)} V(\mathbb{B})}\ \textsc{Lema } 5.0.15 \right)$$

$$\dfrac{\dfrac{\vdots \atop \pi}{\vdots \Big\} V(?d)\text{ veces}}\,\mathrm{T}^{\bullet}\textsc{Delay}}{V(\Gamma) \rhd \bullet^{V(?d)}\,(s' @^{V(?a)}\, u') : \bullet^{V(?d)} \bullet \bullet^{V(?a)} \bullet^{V(?b)}\, V(\mathbb{B})}\,\mathrm{T}^{\bullet}\textsc{Delay}$$

Tomando $t' := \bullet^{V(?d)}\,(s' @^{V(?a)}\, u')$, se tiene que t' es regular pues resulta de aplicar demoras a una aplicación múltiple $s' @^{V(?a)}\, u'$, en la que tanto s' como u' son regulares. Además, tenemos que:

$$\begin{array}{rcll}
(t')^{\circ} & = & (\bullet^{V(?d)}\,(s' @^{V(?a)}\, u'))^{\circ} & \\
 & = & (s' @^{V(?a)}\, u')^{\circ} & \\
 & = & (s')^{\circ}\,(u')^{\circ} & \text{por Observación 5.0.3} \\
 & = & s\,u & \\
 & = & t &
\end{array}$$

Para concluir, basta notar que $\bullet^{V(?d)} \bullet \bullet^{V(?a)} \bullet^{V(?b)}\, V(\mathbb{B}) = \bullet^{V(1+?a+?b+?d)}\, V(\mathbb{B}) = \bullet^{V(?c)}\, V(\mathbb{B}) = V(\bullet^{?c}\,\mathbb{B})$ pues $V \vDash \{1+?a+?b+?d \stackrel{?}{=} ?c\}$ por Observación 5.0.14.

□

Definición 5.0.17 (Valuación V_0). Suponemos que entre todas las incógnitas hay ciertas incógnitas

$$?a_0, ?a_1, ?a_2, ?a_3, \cdots$$

Definimos V_0 como la valuación que cumple que para todo $n \in \mathbb{N}_0$:

$$V_0(?a_n) = n$$

y tal que $V_0(?b)$ toma algún otro valor (irrelevante) cuando $?b$ no es una de dichas incógnitas.

Definición 5.0.18 (Traducción al cálculo con incógnitas). Definimos una traducción $-^{\blacktriangle}$ que dado un tipo (o, respectivamente un contexto de tipado) del cálculo-$\lambda^{\bullet}$ devuelve un tipo (o, respectivamente un contexto de tipado) del cálculo-$\lambda^{?}_{\bullet}$.

1. Para cada tipo A del cálculo-$\lambda^{\bullet}$ definimos un tipo demorado $A^{\blacktriangle}$ del cálculo-$\lambda^{?}_{\bullet}$. Además, para cada tipo A del cálculo-$\lambda^{\bullet}$ que no sea de la forma $\bullet A'$ definimos un tipo (a secas) $A^{\vartriangle}$ del cálculo-$\lambda^{?}_{\bullet}$. Las definiciones de $A^{\blacktriangle}$ y $A^{\vartriangle}$ son mutuamente recursivas, y usan el hecho de que todo tipo A del cálculo-$\lambda^{\bullet}$ se puede escribir de la forma $\bullet^{n} A'$ donde A' no es una demora.

$$\begin{array}{lcll}
(\bullet^{n} A)^{\blacktriangle} & \stackrel{\text{def}}{=} & \bullet^{?a_n}\,(A^{\vartriangle}) & \text{donde } A \text{ no es una demora} \\
A^{\vartriangle} & \stackrel{\text{def}}{=} & \begin{cases} \alpha & \text{si } A = \alpha \\ B^{\blacktriangle} \to C^{\blacktriangle} & \text{si } A = B \to C \end{cases} &
\end{array}$$

2. Sobre contextos de tipado:

$$(x_1 : A_1, \ldots, x_n : A_n)^{\blacktriangle} \stackrel{\text{def}}{=} x_1 : A_1^{\blacktriangle}, \ldots, x_n : A_n^{\blacktriangle}$$

Proposición 5.0.19 (Completitud del cálculo-$\lambda^{?}_{\bullet}$)**.** Sea $t \in \Lambda_\bullet$ un término regular tal que $\Gamma \rhd t : A$. Entonces existe un conjunto de ecuaciones E tal que $\Gamma^\blacktriangle \blacktriangleright_E t^\circ : A^\blacktriangle$ y $V_0 \vDash E$.

Demostración. Procedemos por inducción en t, usando el hecho de que todo término regular t se puede escribir de acuerdo con lo indicado por la Observación 5.0.5.

1. $t = \bullet^n x$. Sabemos $\Gamma \rhd \bullet^n x : A$, luego la situación necesariamente debe ser la siguiente:

$$\cfrac{\cfrac{\cfrac{}{\Gamma', x : B \rhd x : B}\ \mathrm{T^\bullet V{\small AR}}}{\vdots \left.\right\}\, n \text{ veces}}\ \mathrm{T^\bullet D{\small ELAY}}}{\Gamma', x : B \rhd \bullet^n x : \bullet^n B}\ \mathrm{T^\bullet D{\small ELAY}}$$

donde $\Gamma = \Gamma', x : B$ y $A = \bullet^n B$. Sea C tal que $B = \bullet^k C$ y C no es una demora, tenemos:

$$\begin{array}{rcl} A^\blacktriangle & = & (\bullet^n B)^\blacktriangle = (\bullet^n \bullet^k C)^\blacktriangle = \bullet^{?a_{n+k}} C^\triangle \\ \Gamma^\blacktriangle & = & (\Gamma', x : B)^\blacktriangle = (\Gamma')^\blacktriangle, x : \bullet^{?a_k} C^\triangle \\ t^\circ & = & (\bullet^n x)^\circ = x \end{array}$$

Luego,

$$\cfrac{E = \mathbf{Eq}(C^\triangle, C^\triangle)}{(\Gamma')^\blacktriangle, x : \bullet^{?a_k} C^\triangle \blacktriangleright_{E \cup \{?a_{n+k} \stackrel{?}{=} ?a_k + ?a_n\}} x : \bullet^{?a_{n+k}} C^\triangle}\ \mathrm{T^{?}_{\bullet}V{\small AR}}$$

Notar que $V_0 \vDash \mathbf{Eq}(C^\triangle, C^\triangle) \cup \{?a_{n+k} \stackrel{?}{=} ?a_k + ?a_n\}$

2. $t = \bullet^n \lambda x.\, s$ con s regular. Sabemos $\Gamma \rhd \bullet^n \lambda x.\, s : A$, luego la situación es la siguiente:

$$\cfrac{\cfrac{\cfrac{\Gamma, x : B \rhd s : C}{\Gamma \rhd \lambda x.\, s : B \to C}\ \mathrm{T^\bullet A{\small BS}}}{\vdots \left.\right\}\, n \text{ veces}}\ \mathrm{T^\bullet D{\small ELAY}}}{\Gamma \rhd \bullet^n \lambda x.\, s : \bullet^n (B \to C)}\ \mathrm{T^\bullet D{\small ELAY}}$$

donde $A = \bullet^n (B \to C)$. Tenemos además,

$$\begin{array}{rcl} A^\blacktriangle & = & (\bullet^n (B \to C))^\blacktriangle = \bullet^{?a_n} (B^\blacktriangle \to C^\blacktriangle) \\ (\bullet^n \lambda x.\, s)^\circ & = & \lambda x.\, s^\circ \end{array}$$

Como vale $\Gamma, x : B \rhd s : C$, por HI tenemos:

$$\Gamma^\blacktriangle, x : B^\blacktriangle \blacktriangleright_E s^\circ : C^\blacktriangle \qquad \text{donde } V_0 \vDash E$$

Luego,

$$\cfrac{\cfrac{\text{HI}}{\Gamma^\blacktriangle, x : B^\blacktriangle \blacktriangleright_E s^\circ : C^\blacktriangle}}{\Gamma^\blacktriangle \blacktriangleright_E \lambda x.\, s^\circ : \bullet^{?a_n} (B^\blacktriangle \to C^\blacktriangle)}\ \mathrm{T^{?}_{\bullet}A{\small BS}}$$

3. $t = \bullet^n (s \,@^k\, u)$ con s y u regulares. Sabemos $\Gamma \rhd \bullet^n (s \,@^k\, u) : A$, luego la situación es la siguiente:

$$\dfrac{\dfrac{\dfrac{\Gamma \rhd s : \bullet^k (C \to B) \quad \Gamma \rhd u : C}{\Gamma \rhd s \,@^k\, u : \bullet^{k+1} B}\ \textsc{Lema 5.0.15}}{\vdots \left.\right\} n \text{ veces}}\ \mathrm{T}^\bullet\textsc{Delay}}{\Gamma \rhd \bullet^n (s \,@^k\, u) : \bullet^n \bullet^{k+1} B}\ \mathrm{T}^\bullet\textsc{Delay}$$

donde $A = \bullet^n \bullet^{k+1} B$. Sea D tal que $B = \bullet^j D$ y D no es una demora, tenemos:

$$\begin{array}{lcl} A^\blacktriangle & = & (\bullet^n \bullet^{k+1} B)^\blacktriangle = (\bullet^n \bullet^{k+1} \bullet^j D)^\blacktriangle = \bullet^{?a_{n+k+1+j}} D^\triangle \\ B^\blacktriangle & = & (\bullet^j D)^\blacktriangle = \bullet^{?a_j} D^\triangle \\ (\bullet^n (s \,@^k\, u))^\circ & = & s^\circ\, u^\circ \end{array}$$

Como vale $\Gamma \rhd s : \bullet^k (C \to \bullet^j D)$, por HI existe E_1 tal que:

$$\Gamma^\blacktriangle \blacktriangleright_{E_1} s^\circ : \bullet^{?a_k} (C^\blacktriangle \to \bullet^{?a_j} D^\triangle) \quad \text{donde } V_0 \vDash E_1$$

Como vale $\Gamma \rhd u : C$, por HI existe E_2 tal que:

$$\Gamma^\blacktriangle \blacktriangleright_{E_2} u^\circ : C^\blacktriangle \quad \text{donde } V_0 \vDash E_2$$

Luego,

$$\dfrac{\dfrac{\text{HI}}{\Gamma^\blacktriangle \blacktriangleright_{E_1} s^\circ : \bullet^{?a_k} (C^\blacktriangle \to \bullet^{?a_j} D^\triangle)} \qquad \dfrac{\text{HI}}{\Gamma^\blacktriangle \blacktriangleright_{E_2} u^\circ : C^\blacktriangle} \qquad E_3}{\Gamma^\blacktriangle \blacktriangleright_{E_1 \cup E_2 \cup E_3 \cup \{?a_{n+k+1+j} \overset{?}{=} 1 + ?a_k + ?a_j + ?a_n\}} s^\circ\, u^\circ : \bullet^{?a_{n+k+1+j}} D^\triangle}\ \mathrm{T}^?_\bullet\textsc{App}$$

Donde $E_3 = \mathbf{Eq}(C^\blacktriangle, C^\blacktriangle)$.

Notar que $V_0 \vDash E_1 \cup E_2 \cup E_3 \cup \{?a_{n+k+1+j} \overset{?}{=} 1 + ?a_k + ?a_j + ?a_n\}$

□

Corolario 5.0.20. Sea $t \in \Lambda$. Son equivalentes:

1. Existen Γ, A y un conjunto de ecuaciones soluble E tales que $\Gamma \blacktriangleright_E t : A$.

2. t es fuertemente temporizable.

Demostración. Es una consecuencia inmediata de Proposición 5.0.16 y Proposición 5.0.19. □

Teorema 5.0.21. Hay términos que no son fuertemente temporizables.
En efecto, el término $t = (\lambda f.\, f\,(f\,z))\,\mathtt{I}$, no es fuertemente temporizable.

Demostración. Si t fuera fuertemente temporizable, por Corolario 5.0.20, tendríamos que para ciertos Γ, A y E soluble vale:

$$\Gamma \blacktriangleright_E (\lambda f.\, f\,(f\,z))\,\mathtt{I} : A$$

Veamos primero que $t \in \Lambda^{\to}$, es decir, es un término tipable del cálculo-λ. Enumeramos todos los juicios que aparecen en la derivación con números desde (1) hasta (9).

$$\pi_1 = \left(\dfrac{}{(4)\, z : A, f : A \to A \vdash f : A \to A}\,\textsc{TVar} \right)$$

$$\pi_2 = \left(\dfrac{}{(1)\, z : A, f : A \to A \vdash z : A}\,\textsc{TVar} \right)$$

$$\pi_3 = \left(\dfrac{\dfrac{}{(2)\, z : A, f : A \to A \vdash f : A \to A}\,\textsc{TVar} \qquad \begin{array}{c}\vdots\\ \pi_2\end{array}}{(3)\, z : A, f : A \to A \vdash f\, z : A}\,\textsc{TApp} \right)$$

$$\pi_4 = \left(\dfrac{\dfrac{}{(6)\, z : A, x : A \vdash x : A}\,\textsc{TVar}}{(7)\, z : A \vdash \lambda x.\, x : A \to A}\,\textsc{TAbs} \right)$$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_1\end{array} \qquad \begin{array}{c}\vdots\\ \pi_3\end{array}}{(5)\, z : A, f : A \to A \vdash f\, (f\, z) : A}\,\textsc{TApp}}{(8)\, z : A \vdash \lambda f.\, f\, (f\, z) : (A \to A) \to A}\,\textsc{TAbs} \qquad \begin{array}{c}\vdots\\ \pi_4\end{array}}{(9)\, z : A \vdash (\lambda f.\, f\, (f\, z))\, \mathtt{I} : A}\,\textsc{TApp}$$

De acuerdo con la hipótesis, tendríamos que tener una derivación para el juicio:

$$\Gamma \blacktriangleright_E (\lambda f.\, f\, (f\, z))\, \mathtt{I} : A$$

en el cálculo-$\lambda^?_\bullet$. Si dicha derivación existiera, el árbol tendría que tener la misma forma que el árbol dado arriba en cálculo-λ simplemente tipado, pero usando la regla $\mathrm{T}^?_\bullet\textsc{Var}$ en lugar de la regla $\textsc{TVar}$, la regla $\mathrm{T}^?_\bullet\textsc{Abs}$ en lugar de la regla $\textsc{TAbs}$ y la regla $\mathrm{T}^?_\bullet\textsc{App}$ en lugar de la regla $\textsc{TApp}$.

Vamos a partir de la derivación en cálculo-λ simplemente tipado para determinar cuáles son las ecuaciones que deberían verificarse para que pueda existir una derivación en cálculo-$\lambda^?_\bullet$. Por último, vamos a argumentar que dichas ecuaciones no se pueden satisfacer.

Como convención de notación, para evitar introducir incógnitas auxiliares, en el caso de la regla $\mathrm{T}^?_\bullet\textsc{Var}$ vamos a escribir una desigualdad $?a \leq ?b$ en lugar de una ecuación $?b \stackrel{?}{=} ?a + ?c$ que involucra a una incógnita $?c$ fresca. De la misma forma, para la regla $\mathrm{T}^?_\bullet\textsc{App}$ vamos a escribir $1 + ?a + ?b \leq ?c$ en lugar de $?c \stackrel{?}{=} 1 + ?a + ?b + ?d$.

Además, en algunos casos, en lugar de escribir dos tipos potencialmente distintos A_1 y A_2 manteniendo registro de un conjunto de ecuaciones $\mathbf{Eq}(A_1, A_2)$ que los iguala, vamos a escribir directamente un único tipo A y a ignorar dichas ecuaciones.

Empecemos mirando el juicio (1):

$$(1)\, z : A, f : A \to A \vdash z : A$$

En cálculo-$\lambda^?_\bullet$ vamos a tener por la regla $\mathrm{T}^?_\bullet\textsc{Var}$ un juicio de la siguiente forma:

$$(1)\, z : \bullet^{?a} A, f : \ldots \blacktriangleright_{E_1} z : \bullet^{?a'} A \quad \text{donde } E_1 = \{?a \leq ?a'\}$$

Notemos que solo con esto no sabemos el tipo de f en el contexto pero podemos de la misma forma verlo con (2) y (3), por las reglas $\mathrm{T}^?_\bullet\textsc{App}$ y $\mathrm{T}^?_\bullet\textsc{Var}$:

$$\dfrac{\dfrac{}{(2)\,\Gamma \blacktriangleright_{E_2} f : \bullet^{?b'}(\bullet^{?a'} A \to \bullet^{?d} A)}\,\mathrm{T}^?_\bullet\textsc{Var} \qquad \dfrac{}{(1)\,\Gamma \blacktriangleright_{E_1} z : \bullet^{?a'} A}\,\mathrm{T}^?_\bullet\textsc{Var}}{(3)\, z : \bullet^{?a} A, f : \bullet^{?b}(\bullet^{?a'} A \to \bullet^{?d} A) \blacktriangleright_{E_3} f\, z : \bullet^{?c} A}\,\mathrm{T}^?_\bullet\textsc{App}$$

Donde

$$\begin{array}{lcl} \Gamma & = & z : \bullet^{?a} A, f : \bullet^{?b}(\bullet^{?a'} A \to \bullet^{?d} A) \\ E_1 & := & \{?a \leq ?a'\} \\ E_2 & := & \{?b \leq ?b'\} \\ E_3 & := & E_1 \cup E_2 \cup \{1+?b'+?d \leq ?c\} \end{array}$$

Notemos que el contexto de tipado es siempre el mismo. Sigamos ahora con (4) y (5) usando las reglas $\mathrm{T}^?_\bullet\textsc{Var}$ y $\mathrm{T}^?_\bullet\textsc{App}$:

$$\dfrac{\dfrac{}{(4)\,\Gamma \blacktriangleright_{E_4} f : \bullet^{?b''}(\bullet^{?a'} A \to \bullet^{?d} A)}\,\mathrm{T}^?_\bullet\textsc{Var} \qquad \begin{array}{c}\vdots\\(3)\end{array}}{(5)\,\Gamma \blacktriangleright_{E_5} f\,(f\,z) : \bullet^{?e} A}\,\mathrm{T}^?_\bullet\textsc{App}$$

Donde

$$\begin{array}{lcl} E_4 & := & \{?b \leq ?b''\} \\ E_5 & := & E_4 \cup E_3 \cup \mathbf{Eq}(\bullet^{?a'} A, \bullet^{?c} A) \cup \{1+?b''+?d \leq ?e\} \\ & = & E_4 \cup E_3 \cup \{?a' \stackrel{?}{=} ?c, 1+?b''+?d \leq ?e\} \end{array}$$

Continuamos con (6) y (7) usando las reglas $\mathrm{T}^?_\bullet\textsc{Var}$ y $\mathrm{T}^?_\bullet\textsc{Abs}$, tenemos que:

$$\dfrac{\dfrac{}{(6)\,\Gamma, x : \bullet^{?h} A \blacktriangleright_{E_6} x : \bullet^{?h'} A}\,\mathrm{T}^?_\bullet\textsc{Var}}{(7)\,\Gamma \blacktriangleright_{E_7} \lambda x.\, x : \bullet^{?i}(\bullet^{?h} A \to \bullet^{?h'} A)}\,\mathrm{T}^?_\bullet\textsc{Abs}$$

Donde

$$\begin{array}{lcl} E_6 & := & \{?h \leq ?h'\} \\ E_7 & := & E_6 \end{array}$$

Finalmente, analizando (8) y (9) con las reglas $\mathrm{T}^?_\bullet\textsc{Abs}$ y $\mathrm{T}^?_\bullet\textsc{App}$ tenemos que:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\(5)\end{array}}{(8)\,\Gamma \blacktriangleright_{E_8} \lambda f.\, f\,(f\,z) : \bullet^{?g}(\bullet^{?b}(\bullet^{?a'} A \to \bullet^{?d} A) \to \bullet^{?e} A)}\,\mathrm{T}^?_\bullet\textsc{Abs} \qquad \begin{array}{c}\vdots\\(7)\end{array}}{(9)\,\Gamma \blacktriangleright_{E_9} (\lambda f.\, f\,(f\,z))\,\mathtt{I} : \bullet^{?j} A}\,\mathrm{T}^?_\bullet\textsc{App}$$

Donde

$$\begin{array}{lcl} E_8 & := & E_5 \\ E_9 & := & E_8 \cup E_7 \cup \mathbf{Eq}(\bullet^{?b}(\bullet^{?a'} A \to \bullet^{?d} A), \bullet^{?i}(\bullet^{?h} A \to \bullet^{?h'} A)) \cup \{1+?g+?e \leq ?j\} \\ & = & E_8 \cup E_7 \cup \{?b \stackrel{?}{=} ?i, ?a' \stackrel{?}{=} ?h, ?d \stackrel{?}{=} ?h', 1+?g+?e \leq ?j\} \end{array}$$

Resumiendo, tenemos que las ecuaciones que se generan en cada paso son las siguientes:

1. $\{?a \leq ?a'\}$
2. $\{?b \leq ?b'\}$
3. $\{1+?b'+?d \leq ?c\}$
4. $\{?b \leq ?b''\}$
5. $\{?a' \stackrel{?}{=} ?c, 1+?b''+?d \leq ?e\}$
6. $\{?h \leq ?h'\}$
7. $\varnothing$
8. $\varnothing$
9. $\{?b \stackrel{?}{=} ?i, ?a' \stackrel{?}{=} ?h, ?d \stackrel{?}{=} ?h', 1+?g+?e \leq ?j\}$

Para que el juicio $\Gamma \blacktriangleright_{E_9} (\lambda f.\, f\,(f\,z))\,\mathtt{I} : A$ sea tipable en el cálculo-$\lambda^?_\bullet$, debería existir una valuación que haga verdaderas a todas las ecuaciones de arriba. A continuación, vamos a argumentar que esto no es posible.

Por (6) tenemos $?h \leq ?h'$. Por lo tanto debería valer lo siguiente:

$$\begin{array}{rcll} ?h \leq ?h' & \iff & ?a' \leq ?h' & \text{pues } ?a' = ?h \text{ por (9)} \\ & \iff & ?a' \leq ?d & \text{pues } ?h = ?d \text{ por (9)} \\ & \iff & ?c \leq ?d & \text{pues } ?a' = ?c \text{ por (5)} \end{array}$$

Observar que no es posible que $?c \leq ?d$ pues $\{1+?b'+?d \leq ?c\}$ por (3). Por lo tanto no existe una valuación V tal que $V \vDash E_9$ como para que valga el juicio:

$$\Gamma \blacktriangleright_{E_9} (\lambda f.\, f\,(f\,z))\,\mathtt{I} : \bullet^{?j} A$$

Por lo tanto, $(\lambda f.\, f\,(f\,z))\,\mathtt{I}$ no es fuertemente temporizable. $\square$

6. CONCLUSIÓN

En este trabajo presentamos el cálculo-$\lambda^\bullet$, una extensión del cálculo-λ que nos permite representar el costo de realizar un cómputo. Definimos su sintaxis y su semántica, damos el conjunto de términos, reglas de reducción y el sistema de tipos con sus reglas de tipado. Además, caracterizamos las formas normales y estudiamos propiedades como la confluencia y normalización fuerte. Este cálculo, en particular, hace posible estudiar propiedades de normalización basadas en simples operaciones aritméticas. Por otro lado, también analizamos la posibilidad de "temporizar" términos del cálculo-λ, para lo cual haciendo uso de un sistema auxiliar (el cálculo-$\lambda^?_\bullet$) vemos que hay términos del cálculo-λ que no son fuertemente temporizables.

En lo que resta de este capítulo, mencionamos primero, en la Sección 6.1, algunos trabajos relacionados con este y planteamos preguntas que quedan de trabajo futuro en relación a estos. Luego, en la Sección 6.2 esbozamos un sistema concreto que sería interesante estudiar en el futuro, en particular una extensión del cálculo-$\lambda^\bullet$ con una noción de subtipado.

6.1. Trabajo relacionado

En esta sección comparamos el trabajo hecho sobre el cálculo-$\lambda^\bullet$ con dos trabajos relacionados. Mencionamos sus similitudes y diferencias y planteamos ciertas preguntas que permitirían potencialmente conectar estos trabajos con el presentado en esta tesis.

6.1.1. *Clocked Lambda Calculus*

El trabajo de Endrullis *et al.* [12] sobre un cálculo-λ cronometrado plantea un cálculo sin tipos con reglas similares a las planteadas en este trabajo. Las reglas son las siguientes:

$$\begin{array}{rcl} (\lambda x.\, M)\, N & \to & \tau(M\{x := N\}) \\ \tau(M)\, N & \to & \tau(M\, N) \end{array}$$

En este caso, se representa la espera con τ en lugar de $\bullet$. Podemos ver que la primera regla es exactamente la misma a la dada por la relación $\to_{\beta\bullet}$ y la segunda cumple una función similar a la de la regla que denominamos $\to_w$ en este trabajo. El trabajo de Endrullis estudia propiedades de *reescritura infinitaria*, como la confluencia, la normalización fuerte y la unicidad de formas normales, sobre el cálculo que plantea. Dado que el cálculo-$\lambda^\bullet$ tiene reglas parecidas, es natural preguntarse si cumple estas mismas propiedades.

Por otro lado, el interés del trabajo de [12] es que el cálculo-λ cronometrado que definen sirve para discriminar términos de manera más refinada a través de sus *Böhm trees* (BT). Más precisamente, algunos términos que tienen el mismo BT en el cálculo-λ, resultan tener distintos BTs en el cálculo-λ cronometrado. Esta es otra propiedad que posiblemente se podría estudiar en el marco del cálculo-$\lambda^\bullet$.

6.1.2. *Exact bounds for lengths of reductions in typed lambda-calculus*

El trabajo de Beckmann [13] en el que se establecen cotas exactas para la longitud de la reducción de λ-términos determina la siguiente cota superior de reducción:

$$dl_n(N) \leq 2_n(N)$$

donde la función $2_n(N)$ corresponde a una torre de exponenciales:

$$2_0(n) = n \quad 2_{m+1}(n) = 2^{2_m(n)}$$

mientras que $dl_n(N)$ denota la longitud de la reducción más larga que comienza en un término de tamaño a lo sumo N y de *grado* a lo sumo n. Más precisamente, se define:

$$dl_n(N) \stackrel{\text{def}}{=} \text{máx}\{d(t) \mid g(t) \leq n, \ |t| \leq N\}$$

donde:

- t varía sobre todos los posibles términos,
- $g(t)$ denota el *grado* del término t, definido como el orden del tipo más grande de cualquiera de los subtérminos de t,
- $d(t)$ denota la longitud de la reducción más larga que comienza en t.

Esta cota, al igual que la presentada en este trabajo (Teorema 4.2.16) , se obtiene a partir de simples cálculos aritméticos. Sin embargo, hay dos diferencias importantes. En primer lugar, el trabajo de Beckmann da una cota para todas las posibles reducciones a forma normal, mientras que en nuestro caso la función de crecimiento da una cota para *una* reducción en particular. En segundo lugar, la cota que proponemos en este trabajo se achica después de un paso de reducción, lo cual sirve para demostrar normalización débil, en tanto que en el trabajo de Beckmann no hay ninguna garantía de que la cota se achique.

Queda como trabajo futuro evaluar la posibilidad de usar el cálculo-$\lambda^\bullet$ para conseguir una mejor cota, similar en orden de complejidad a la del trabajo de Beckmann, pero que además sirva para probar normalización débil.

6.1.3. *The Clocks Are Ticking: No More Delays!*

El trabajo de Bahr *et al.* [14] presenta una teoría cronometrada, agregando relojes y ticks de reloj al cálculo-λ. Para esto agrega, al igual que en este trabajo, una demora notada $\triangleright A$ para reflejar que A estará listo después de un tiempo. A diferencia de este trabajo, agrega también un contexto para los relojes y nuevas reglas para plasmar el paso del tiempo. Quedaría ver si algo de esto puedo ser aplicado al cálculo-$\lambda^\bullet$.

6.2. Trabajo futuro – Extensión con subtipado

Como ya vimos, hay términos tipables en cálculo-λ que no son fuertemente temporizables. Esto limita la expresividad del cálculo-$\lambda^\bullet$. Una intuición es que esto se debe a que una misma función $f : A \to B$ se puede usar más de una vez, aplicándola a argumentos de tipo A que demoren distinta cantidad de tiempo en producir un resultado.

Una posible manera de aumentar la expresividad del cálculo-$\lambda^\bullet$ es permitiendo que un término de tipo A pueda ser usado también como un término de tipo $\bullet A$. Por ejemplo, si una función recibe un argumento que produce un dato después de k unidades de tiempo, debería ser posible usarla pasándole un dato que requiera esperar a lo sumo k unidades de tiempo.

Más precisamente, se podría considerar una relación de subtipado $A <: B$ que incluya las siguientes reglas:

$$\frac{\Gamma \rhd t : A \quad A <: B}{\Gamma \rhd t : B}\ \mathrm{T^\bullet S{\scriptstyle UB}} \qquad \frac{}{A <: \bullet A}\ \mathrm{D{\scriptstyle ELAY}}$$

Además, habría que agregar las reglas usuales de reflexividad, transitividad y congruencia para la relación de subtipado:

$$\frac{}{A <: A}\ \mathrm{R{\scriptstyle EFL}} \qquad \frac{A <: B \quad B <: C}{A <: C}\ \mathrm{T{\scriptstyle RANS}}$$

$$\frac{A' <: A \quad B <: B'}{A \to B <: A' \to B'}\ \mathrm{C{\scriptstyle ONG}A{\scriptstyle RROW}} \qquad \frac{A <: A'}{\bullet A <: \bullet A'}\ \mathrm{C{\scriptstyle ONG}D{\scriptstyle ELAY}}$$

Queda pendiente verificar si el sistema así extendido sigue teniendo las buenas propiedades ya estudiadas para el cálculo-$\lambda^\bullet$, así como estudiar en qué medida esta extensión aumenta la expresividad.

Bibliografía

[1] Alonzo Church. *The Calculi of Lambda-conversion*. Annals of mathematics studies. Princeton University Press, 1941.

[2] Hendrik Pieter Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1984.

[3] Peter Selinger. Lecture notes on the lambda calculus. *CoRR*, abs/0804.3434, 2008.

[4] Gerald Jay Sussman and Guy L Jr. Steele. Scheme: An interpreter for extended lambda calculus. In *MEMO 349, MIT AI LAB*, 1975.

[5] Simon L. Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice-Hall international series in computer science. Prentice/Hill International, 1987.

[6] Andrew W. Appel. *Compiling with Continuations*. Cambridge University Press, 2007.

[7] Hendrik Pieter Barendregt, Will Dekkers, and Richard Statman. *Lambda Calculus with Types*. Lambda Calculus with Types. Cambridge University Press, 2013.

[8] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Term Rewriting and All that. Cambridge University Press, 1998.

[9] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[10] Benjamin C. Pierce. *Types and Programming Languages*. The MIT Press. MIT Press, 2002.

[11] Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism*. Number v. 10 in Lectures on the Curry-Howard isomorphism. Elsevier, 2006.

[12] Jörg Endrullis, Dimitri Hendriks, Jan Willem Klop, and Andrew Polonsky. Clocked lambda calculus. *Math. Struct. Comput. Sci.*, 27(5):782–806, 2017.

[13] Arnold Beckmann. Exact bounds for lengths of reductions in typed lambda-calculus. *J. Symb. Log.*, 66(3):1277–1285, 2001.

[14] Patrick Bahr, Hans Bugge Grathwohl, and Rasmus Ejlers Møgelberg. The clocks are ticking: No more delays! In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2017.