Available online at www.sciencedirect.com

SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 143 (2006) 45–57

www.elsevier.com/locate/entcs

Lowness Properties and Approximations of the Jump

Santiago Figueira[1,4]

Department of Computer Science, FCEyN
University of Buenos Aires
Argentina

André Nies[2]

Department of Computer Science
University of Auckland
New Zealand

Frank Stephan[3,5]

Departments of Computer Science and Mathematics
National University of Singapore
Singapore

Abstract

We study and compare two combinatorial lowness notions: *strong jump-traceability* and *well-approximability of the jump*, by strengthening the notion of jump-traceability and ω-r.e. for sets of natural numbers. We prove that there is a strongly jump-traceable set which is not computable, and that if A' is well-approximable then A is strongly jump-traceable. For r.e. sets, the converse holds as well. We characterize jump-traceability and the corresponding strong variant in terms of Kolmogorov complexity, and we investigate other properties of these lowness notions.

Keywords: Lowness, traceability, ω-r.e., K-triviality, Kolmogorov complexity

[1] Email: sfigueir@dc.uba.ar

[2] Email: andre@cs.auckland.ac.nz

[3] Email: fstephan@comp.nus.edu.sg

[4] S. Figueira is partially supported by a grant of Fundación Antorchas.

[5] F. Stephan is partially supported by NUS grant number R252–000–212–112.

1571-0661
doi:10.1016/j.entcs.2005.05.025

1 Introduction

A *lowness property* of a set A says that A is computational weak when used as an oracle, and hence A is close to being computable. In this article we study and compare some "combinatorial" lowness properties in the direction of characterizing K-trivial sets.

A set is K-trivial when it is highly compressible in terms of Kolmogorov complexity (see Section 2 for the formal definition). In [8], Nies proved that a set is K-trivial if and only if A is low for Martin-Löf-random (i.e. each Martin-Löf-random set is already random relative to A).

Terwijn and Zambella [12] defined a set A to be *recursively traceable* if there is a recursive bound p such that for every $f \leq_T A$, there is a recursive r such that for all x, $|D_{r(x)}| \leq p(x)$, and $(D_{r(x)})_{x\in\mathbb{N}}$ is a set of possible values of f: for all x, we have and $f(x) \in D_{r(x)}$. They showed that this combinatorial notion characterizes the sets that are low for Schnorr tests.

This property was modified in [9] to *jump-traceability*. A set A is jump traceable if its jump at argument e, written $J^A(e) = \{e\}^A(e)$, has few possible values.

Definition 1.1 A uniformly r.e. family $T = \{T_0, T_1, \ldots\}$ of sets of natural numbers is a *trace* if there is a recursive function h such that $\forall n\ |T_n| \leq h(n)$. We say that h is a *bound* for T. The set A is *jump-traceable* if there is a trace T such that $\forall e\ [J^A(e)\downarrow \Rightarrow\ J^A(e) \in T_e]$. We say that A is jump traceable *via a function* h if, additionally, T has bound h.

Another notion studied in [9] is *super-lowness*, first introduced in [2,7].

Definition 1.2 A set A is *ω-r.e.* iff there exists a recursive function b such that $A(x) = \lim_{s\to\infty} g(x,s)$ for a recursive $\{0,1\}$-valued g such that $g(x,s)$ changes at most $b(x)$ times. In this case, we say that A is ω-r.e. *via the function* g and *bound* b. A is *super-low* iff A' is ω-r.e.

Both jump-traceable and super-low sets are closed downward under Turing reducibility and imply being generalized low (i.e. $A' \leq A \oplus \emptyset'$). In [9] jump-traceability and super-lowness were studied and compared, proving that these two lowness notions coincide within the r.e. sets but that none of them implies the other within the ω-r.e. sets.

In this article, we define the notions of *strong jump-traceability* (see Definition 3.2) and *well-approximability* (see Definition 4.1), by strengthening the notions of jump-traceability and ω-r.e., respectively. A special emphasis is given to the case where the jump of A is ω-r.e. The strong variant of these notions consider *all* orders as the bound instead of just *some* recursive bound.

Here, an *order* is a slowly growing but unbounded recursive function (see Definition 3.1). Among our main results are:

- There is a non-computable strongly jump-traceable set;
- If A' is well-approximable then A is strongly jump-traceable. The converse also holds, if A is r.e.

Our approach is used to study interesting lowness properties related to plain and prefix-free Kolmogorov complexity. We investigate the properties of sets A such that the Kolmogorov complexity relative to A is only a bit smaller than the unrelativized one. We prove some characterizations of jump-traceability and its strong variant in terms of prefix-free (denoted with K) and plain (denoted with C) Kolmogorov complexity, respectively:

- A is jump-traceable if and only if there is a recursive p, growing faster than linearly such that $K(y)$ is bounded by $p(K^A(y)+c_0)+c_1$, for some constants c_0 and c_1;
- A is strongly jump-traceable if and only if $C(x) - C^A(x)$ is bounded by $h(C^A(x))$, for every order h and almost all x.

We know that K-triviality implies jump-traceability, but it is unknown whether K-triviality implies strong jump-traceability. The reverse direction is also open.

2 Basic definitions

If A is a set of natural numbers then $A(x) = 1$ if $x \in A$; otherwise $A(x) = 0$. We denote with $A \upharpoonright n$ the string of length n which consists of the bits $A(0) \dots A(n-1)$.

If A is given a Δ^0_2-approximation and Ψ is a functional, we write $\Psi^A(e)[s]$ for $\Psi^{A_s}_s(e)$. From a partial recursive functional Ψ, one can effectively obtain a primitive recursive and strictly increasing function α, called a *reduction function* for Ψ, such that $\forall X \, \forall e \;\, \Psi^X(e) = J^X(\alpha(e))$.

For each real A, we want to define $K^A(y)$ as the length of a shortest prefix-free description of y using oracle A. An *oracle machine* is a partial recursive functional $M : \{0,1\}^\infty \times \{0,1\}^* \mapsto \{0,1\}^*$. We write $M^A(x)$ for $M(A,x)$. M is an *oracle prefix-free machine* if the domain of M^A is an antichain under inclusion of strings, for each A. Let $(M_d)_{d\in\mathbb{N}}$ be an effective listing of all oracle prefix-free machines. The universal oracle prefix-free machine U is given by $U^A(0^d 1\sigma) = M^A_d(\sigma)$ and the prefix-free Kolmogorov complexity relative to A is defined as $K^A(y) = \min\{|\sigma| \colon U^A(\sigma) = y\}$, where $|\sigma|$ denotes the length of σ. If $A = \emptyset$, we simply write $U(\sigma)$ and $K(y)$. As usual, $U(\sigma)[s] \downarrow= y$

indicates that $U(\sigma) = y$ and the computation takes at most s steps. We say that $A \in \{0,1\}^\infty$ is Martin-Löf random iff $\exists c\ \forall n\ K(A \upharpoonright n) > n - c$. A set A is K-trivial iff $\exists c\ \forall n\ K(A \upharpoonright n) \le K(n) + c$.

The Kraft-Chaitin Theorem states that from a computably enumerable sequence of pairs $(\langle n_i, \sigma_i \rangle)_{i\in\mathbb{N}}$ (known as *axioms*) such that $\sum_{i\in\mathbb{N}} 2^{-n_i} \le 1$, we can effectively obtain a prefix-free machine M such that for each i there is a τ_i of length n_i with $M(\tau_i)\downarrow = \sigma_i$, and $M(\rho)\uparrow$ unless $\rho = \tau_i$ for some i.

If we drop the condition of the domain of M^A being an antichain, we obtain a similar notion, called plain Kolmogorov complexity and denoted by C. Hence, $C^A(y)$ will denote the length of the shortest description of y using oracle A, when we do not have the restriction on the domain

A *binary machine* is a partial recursive function $\tilde{M} : \{0,1\}^* \times \{0,1\}^* \mapsto \{0,1\}^*$. Let $\tilde{U}$ be a binary universal function i.e. $\tilde{U}(0^d 1\sigma, x) = \tilde{M}_d(\sigma, x)$, where $(\tilde{M}_d)_{d\in\mathbb{N}}$ is an enumeration of all partial recursive functions of two arguments. We define the plain conditional Kolmogorov complexity $C(y|x)$ as the length of the shortest description of y using $\tilde{U}$ with string x as the second argument, i.e. $C(y|x) = \min\{|\sigma| : \tilde{U}(\sigma, x) = y\}$.

Let $str : \mathbb{N} \to \{0,1\}^*$ be the standard enumeration of the strings. The string $str(n)$ is that binary sequence $b_0 b_1 \dots b_m$ for which the binary number $1 b_0 b_1 \dots b_m$ has the value $n+1$. Thus, $str(0) = \lambda$, $str(1) = 0$, $str(2) = 1$, $str(3) = 00$, $str(4) = 01$ and so on.

3 Strong jump-traceability

Recall that an r.e. set A is *promptly simple* if A is co-infinite and there is a recursive function p and an effective approximation $(A_s)_{s\in\mathbb{N}}$ of A such that, for each e, if $|W_e| = \infty$ then $\exists s\ \exists x\ [x \in W_{e,s} \setminus W_{e,s}\ \wedge\ x \in A_{p(s)} \setminus A_{p(s)-1}]$. In this section, we introduce a stronger version of jump-traceability and we prove that there is a promptly simple (hence non recursive) strongly jump-traceable set. We also prove that there is no maximal order as bound for jump-traceability.

Definition 3.1 A function $h : \mathbb{N} \to \mathbb{N}^+$ is an *order* iff h is recursive, $\forall x\ h(x) \le h(x+1)$ and $\lim_{x\to\infty} h(x) = \infty$.

Notice that any reduction function is an order.

Definition 3.2 A set A is *strongly jump-traceable* iff for each order h, A is jump traceable via h.

Clearly, strong jump-traceability implies jump-traceability and it is not difficult to see that strong jump-traceability is closed downward under Turing reducibility.

Notice that if A is recursive then A is strongly jump-traceable because we can trace the jump by $T_e = \{J^A(e)\}$ if $J^A(e)\downarrow$ and $T_e = \emptyset$ otherwise.

Theorem 3.4 below shows that the converse is not true. To prove it, we need the following Lemma which states that there is a function growing slower than all orders which is recursively approximable from above.

Lemma 3.3 *There exists $g\colon \mathbb{N} \to \mathbb{N}$ such that*

(i) $\forall x\ g(x) = \lim_{s\to\infty} g_s(x)$, *where* $g(s,x) = g_s(x)$ *is recursive and* $g_s(x) \geq g_{s+1}(x)$;

(ii) $\lim_{x\to\infty} g(x) = \infty$;

(iii) *For all orders h, $g(x) \leq h(x)$ for almost all x.*

Proof. Define $G_s(x) = x + \max\{\varphi_{e,s}(y) : \varphi_{e,s}(y)\downarrow\ \wedge e \leq x \wedge y \leq x\}$. Clearly, $G(s,x) = G_s(x)$ is recursive and it is easy to see that for all x, $G_s(x) \leq G_{s+1}(x)$ and for all s, $G_s(x) < G_s(x+1)$. Also $G_s(x) \geq \varphi_{e,s}(x)$ for all $e \leq x$. Let us define $G = \lim_{s\to\infty} G_s$. Then G grows faster than any recursive function, that is, if $\varphi_e(x)$ is defined, then $G(x) \geq \varphi_e(x)$ for all $e \leq x$.

Let us define now the "inverse of G" as follows: $g_s(y) = \max\{x\colon G_s(x) \leq y\}$ if $G_s(0) \leq y$ and $g_s(y) = 0$ otherwise; we also define $g = \lim_{s\to\infty} g_s$. Since G_s is recursive and monotone increasing in x, g_s is recursive and $g_s \geq g_{s+1}$. This proves (i).

Also g is unbounded because G is. Hence, (ii) is satisfied.

For (iii), let h be any order. The function $H(x) = \min\{y\colon h(y) \geq x\}$ is recursive because h is unbounded by hypothesis. Then, there is e such that $H = \varphi_e$. By the construction of G, $\forall x\ [x \geq e \ \Rightarrow\ G(x) \geq H(x)]$. We will prove that $g(y) = \max\{x\colon G(x) \leq y\} \leq h(y)$ for all $y \geq G(e)$ and $g(y) \geq e$. Fix $y \geq G(e)$ and suppose that $x \geq e$ and $G(x) \leq y$. Since h is monotone, $h(G(x)) \leq h(y)$ and since H is below G beyond e, $h(H(x)) \leq h(G(x))$. By the definition of H, $h(H(x)) \geq x$, so finally we obtain $x \leq h(y)$. □

Theorem 3.4 *There exist a promptly simple strongly jump-traceable set.*

Proof. We construct a promptly simple set A in stages satisfying the requirements

$$P_e : |W_e| = \infty \ \Rightarrow\ \exists s \exists x\ [x \in W_{e,s} \setminus W_{e,s-1}\ \wedge\ x \in A_s \setminus A_{s-1}].$$

During the construction, P_e may destroy $J^A(k)$ at stage s only if $e < g_s(k)$.

Construction of A. Let g_s be the one defined in Lemma 3.3.

Stage 0: set $A_0 = \emptyset$.

Stage $s+1$: choose the least $e \leq s$ such that

- P_e yet not satisfied;

- There exists x such that $x \in W_{e,s+1} \setminus W_{e,s}$, $x > 2e$ and for all k such that $g_s(k) \leq e$, if $J^A(k)[s]$ is defined then x is greater than the use of $J^A(k)[s]$.

If such e exists, put least such x for e into A_{s+1}. We say that P_e *receives attention* at stage $s+1$, and declare P_e satisfied. Otherwise, $A_{s+1} = A_s$. Finally, define $A = \bigcup_s A_s$.

Verification. Clearly, P_e receives attention at most once. So we can use below the fact that every requirement influences the enumeration of A at most once.

To show that A is strongly jump-traceable, fix a recursive order h. We will prove that there exists an r.e. trace T for J^A as in Definition 1.1. Let h be any order. By Lemma 3.3, there exists k_0 such that for all $k \geq k_0$, $g(k) \leq h(k)$. Define the recursive function $f(k) = \min\{s\colon g_s(k) \leq h(k)\}$ if $k \geq k_0$ and $f(k) = 0$ otherwise. For $k \geq k_0$ and $s \geq f(k)$, $g_s(k)$ will be below $h(k)$, so $J^A(k)$ may change because P_e receives attention, for $e < g_s(k) \leq h(k)$. Since each P_e receives attention at most once, $J^A(k)$ can change at most $h(k)$ times after stage $f(k)$. So

$$T_k = \begin{cases} \{J^A(k)[s]\colon J^A(k)[s]\downarrow \ \wedge\ s \geq f(k)\} & \text{if } k \geq k_0; \\ \{J^A(k)\} & \text{if } J^A(k)\downarrow \ \wedge\ k < k_0; \\ \emptyset & \text{otherwise.} \end{cases}$$

is as required.

Fix e such that W_e is infinite and let us see that P_e is met. Let s such that $\forall k\ [g(k) \leq e \ \Rightarrow\ g_s(k) = g(k)]$ and $s' > s$ such that no P_i receives attention after stage s' for any $i < e$. Then, by the construction, no computation $J^A(k)$, $g(k) \leq e$ can be destroyed after stage s'. So there is $t > s'$ such that for all k where $g_t(k) \leq e$, if $J^A(k)$ converges then the computation is stable from stage t on. Choose $t' \geq t$ such that there is $x \in W_{e,t'+1} \setminus W_{e,t'}$, $x > 2e$ and x is greater than the use of all converging $J^A(k)$ for all k where $g_{t'}(k) \leq e$. Now either P_e was already satisfied or P_e receives attention at stage $t'+1$. In either case P_e is met. □

We investigate about the existence of a maximal bound for jump-traceability. Given an order h, is it always possible to find a jump-traceable set A for which h is too small to be a bound for any trace for the jump of A? The next Theorem answers this question positively.

Theorem 3.5 *For any order h there is an r.e. set A and an order $\tilde{h}$ such that A is jump-traceable via $\tilde{h}$ but not via h.*

Proof. We will define an auxiliary functional Ψ and we use α, the reduction function for Ψ (i.e. $\Psi^X(e) = J^X(\alpha(e))$ for all X and e), in advance by the Recursion Theorem. At the same time, we will define an r.e. set A and a trace

$\tilde{T}$ for J^A. Finally, we will verify that there is an order $\tilde{h}$ as stated.

Let $T(0), T(1), \ldots$ be an enumeration of all the traces with bound h, so that $T(e) = \{T(e)_0, T(e)_1, \ldots\}$, the e-th such trace, is as in Definition 1.1. Requirement P_e tries to show that J^A is not traceable via the trace $T(e)$ with bound h, that is,

$$P_e : \exists x \; \Psi^A(x) \notin T(e)_{\alpha(x)}$$

and requirement N_e tries to stabilize the jump when it becomes defined, i.e.

$$N_e : [\exists^\infty s \; J^A(e)[s] \downarrow] \;\Rightarrow\; J^A(e) \downarrow .$$

The strategy for a single procedure P_e consists of an initial action and a possible later action.

Initial action at stage $s+1$:

- Choose a new candidate $x_e = \langle e, n\rangle$, where n is the number of times that P_e has been initialized. Define $\Psi^A(x_e)[s+1] = 0$ with large use.

Action at stage $s+1$:

- Let $x_e = \langle e, n\rangle$ be the current candidate. Put y into A_{s+1}, where y is the use of the defined $\Psi^A(x_e)[s]$. Notice that this action will not affect $J^A(i)[s]$ for $i < e$ because of the choice of y;
- Define $\Psi^A(x_e)[s+1] = \Psi^A(x_e)[s] + 1$ with use $y' > y$ and greater than the use of all defined computations of $J^A(i)[s+1]$ for $i < e$.

We say that P_e *requires attention* at stage $s+1$ if $\Psi^A(x_e)[s] \in T(e)_{\alpha(x_e)}[s]$ and we say that N_e *requires attention* at stage $s+1$ if $J^A(e)[s]$ becomes defined for the first time.

We define $\tilde{T} = \{\tilde{T}_0, \tilde{T}_1, \ldots\}$ by stages. The s-th stage of $\tilde{T}_i$ will be denoted by $\tilde{T}_i[s]$. We start with $A_0 = \emptyset$ and $\tilde{T}_i[0] = \emptyset$ for all i. At stage $s+1$ we consider the procedures N_j for $j \le s$ and P_j for $j < s$. We also initialize the new P_s. We look at the least procedure requiring attention in the order $P_0, N_0, \ldots, P_s, N_s$. If there is no one, do nothing. Otherwise, suppose P_e is the first one. We let P_e take action at $s+1$, changing A below the use of $\Psi^A(x_e)[s]$ and redefining $\Psi^A(x_e)[s+1]$ without affecting N_i for $i < e$. We keep the other computations of P_j with the new definition of A, for $j \neq i$ and large use. If N_e is the least procedure requiring attention, there is y such that $J^A(e)[s] \downarrow = y$. We put y into $\tilde{T}_e[s+1]$ and initialize P_j for $e < j \le s$. In this case, we say that N_e *acts*.

Let us prove that P_e is met. Take s such that all $J^A(i)$ are stable for $i < e$. Suppose x_e is the actual candidate of P_e. Since P_e is not going to be initialized again, x_e is the last candidate it picks. Each time $\Psi^A(x_e)[t] \in T(e)_{\alpha(x_e)}[t]$ for

$t > s$, P_e acts and changes the definition of $\Psi^A(x_e)$ to escape from $T(e)_{\alpha(x_e)}$. Since $|T(e)_{\alpha(x_e)}| \leq h(\alpha(x_e))$, there is $s' > s$ such that $T(e)_{\alpha(x_e)}[s'] = T(e)_{\alpha(x_e)}$. By construction, $\Psi^A(x_e)[s'+1] \notin T(e)_{\alpha(x_e)}$ and $\Psi^A(x_e)[s'+1]$ is stable.

We say that N_e is *injured* at stage $s+1$ if we put y into A_{s+1} and y is $\leq$ the use of $J^A(e)[s]$. We define $c_P(k)$ as a bound for the number of initializations of P_r, for $r \leq k$; and define $c_N(k)$ as a bound for the number of injuries to N_r, for $r \leq k$. Since P_0 is initialized just once and makes at most $h(\langle 0, 0\rangle)$ changes in A, $c_P(0) = 1$ and $c_N(0) = h(\langle 0, 0\rangle)$. The number of times that P_{k+1} is initialized is bounded by the number of times that N_r acts, for $r \leq k$, so $c_P(k+1) = c_P(k) + c_N(k)$. Each time N_r is injured, for $r \leq k$ then N_{k+1} may also be injured; additionally, N_{k+1} may be injured each time P_{k+1} changes A. The latter occurs at most $h(\langle k+1, i\rangle)$ for the i-th initialization of P_{k+1}. Hence $c_N(k+1) = 2c_N(k) + \sum_{i \leq c_P(k+1)} h(\langle k+1, i\rangle)$.

Once N_e is not injured anymore, if $J^A(e)\downarrow$ then $J^A(e) \in \tilde{T}_e$. Since the number of changes of $J^A(k)$ is at most the number of injuries to N_e, we define the function $\tilde{h}(e) = c_N(e)$ which is clearly an order and it constitutes a bound for the trace $(\tilde{T}_i)_{i\in\mathbb{N}}$. □

It is still open if there is no minimal bound for jump-traceability, i.e. it is unknown if given an order h there is a set A and an order $\tilde{h}$ such that A is jump-traceable via h but not via $\tilde{h}$.

4 Well-approximability of the jump

We strengthen the notion of super-lowness and study the relationship to strongly jump-traceable.

Definition 4.1 A set A is *well-approximable* iff for each order b, A is ω-r.e. via b.

Clearly, if A' is well-approximable, then A is super low and it is not difficult to see that well-approximability is closed downward under Turing reducibility. We next prove that if A is r.e. then A is strongly jump-traceable iff A' is well-approximable. We first need the following lemmas.

Lemma 4.2 *Let f and $\hat{f}$ be orders such that $f(x) \leq \hat{f}(x)$ for almost all x.*

(i) *If A is jump-traceable via f then A is jump traceable via $\hat{f}$;*

(ii) *If A is well-approximable via f then A is well-approximable via $\hat{f}$.*

Lemma 4.3 *There exists a recursive γ such that for all r.e. A:*

(i) *If A is jump-traceable via an order h then A is super-low via the order $b(x) = 2h(\gamma(x)) + 2$;*

(ii) *If A is super-low via an order b then A is jump-traceable via the order* $h(x) = \lfloor \frac{1}{2} b(\gamma(x)) \rfloor$.

Proof. Follow the proof of [9, Theorem 4.1], together with Lemma 4.3. □

Theorem 4.4 *Let A be an r.e. set. Then the following are equivalent:*

(i) *A is strongly jump-traceable;*

(ii) *A' is well-approximable.*

Proof. (i)⇒(ii). Given an order b, let us prove that A is super-low via b. By part i of Lemma 4.3, it suffices to define an order h such that $2h(\gamma(x)) + 2 \leq b(x)$ for almost all x. If $b(x) \geq 4$ then define $h(\gamma(x)) = \lfloor \frac{b(x)-2}{2} \rfloor$ and if $b(x) < 4$, define $h(\gamma(x)) = 1$. Since γ can be taken strictly monotone, the above definition is correct and we can complete it to make h an order.

(ii)⇒(i). Given an order h, we will prove that A is jump-traceable via h. By part ii of Lemma 4.3, it suffices to define an order b such that $\lfloor \frac{1}{2} b(\gamma(x)) \rfloor \leq h(x)$ for almost all x. The argument is similar to the previous case. □

Later, in Corollary 5.4, we will improve this result and we will see that, in fact, the implication (ii)⇒(i) holds for any A.

We finish this section by proving that the prefixes $A \upharpoonright n$ of a well-approximable set A have low Kolmogorov complexity of order logarithmic in n. Hence A is not Martin-Löf random and furthermore, the effective Hausdorff dimension is 0. The latter is just equivalent of saying that there is no $c > 0$ such that cn is a linear lower bound for the prefix-free Kolmogorov complexity of $A \upharpoonright n$ for almost all n.

Theorem 4.5 *If A is well-approximable then for almost all n, $K(A \upharpoonright n) \leq 4|n|$.*

Proof. Suppose $A(n) = \lim_{s\to\infty} g(n, s)$, where g is recursive and changes at most n times. Given n, there is a unique s and some $m < n$ such that $g(m, s) \neq g(m + 1, s)$ but $g(q, t) = g(q, t + 1)$ for all $t > s$ and $q < n$. That is, s is the time when g converges on below n and m is the place where the last change takes place. The stage s can be computed from m and the number k of stages with $g(m, t + 1) \neq g(m, t)$. So one can compute $A \upharpoonright n$ from m, n, k. Since $k, m \leq n$, one can, for almost all n, code m, n, k in a prefix-free way in $4|n|$ many bits. This is done by using a prefix of the form 1^q0 followed by $2q$ bits representing n, $2q$ bits representing m and $2q$ bits representing k as binary numbers; here q is just the smallest number such that $2q$ bits are enough. Since $k, m \leq n$ and since $2q \leq |n| + c$ for some constant c and since the additionally necessary coding needed to transform the above representation into a program for U is bounded by a constant, we have that there is a constant d such that

$\forall n\ K(A \upharpoonright n) \leq 3|n| + |n|/2 + d$ and then the relation $K(A \upharpoonright n) \leq 4|n|$ holds for almost all n. In fact, using binary notation to store q instead of 1^q0, it would even give $K(A \upharpoonright n) \leq 3(|n| + \log(|n|))$ for almost all n. □

5 Traceability and plain Kolmogorov complexity

We give a characterization of strong jump-traceability in terms of plain Kolmogorov complexity and we show that if A' is well-approximable then A is strongly jump-traceable for any set A.

Theorem 5.1 *If A' is well-approximable then for every order h and almost all x, $C(x) \leq C^A(x) + h(C^A(x))$.*

Proof. For any function f, let define $\hat{f}(y) = y + f(y)$ for all y. Let $\Psi^A(m, n, q)$ be a functional which does the following:

(i) Compute $x = U^A(q)$. If $U^A(q) \uparrow$ then $\Psi^A(m, n, q) \uparrow$;

(ii) Find the first program p such that $|p| = n$ and $\tilde{U}(p, q) = x$. If there is no such p then $\Psi^A(m, n, q) \uparrow$;

(iii) In case $m \notin [1, n]$ then $\Psi^A(m, n, q) \uparrow$. Otherwise, if the m-th bit of p is 1 then $\Psi^A(m, n, q) \downarrow$, else $\Psi^A(m, n, q) \uparrow$.

Let α be a reduction function such that $J^A(\alpha(m, n, q)) = \Psi^A(m, n, q)$ and let h_0 be any order. Since $h = \lfloor h_0/2 \rfloor$ is also an order, it is sufficient to show that there is a constant c with $C(x) \leq \hat{h}(C^A(x)) + c$ for almost all x, since this will imply that $C(x) \leq \hat{h}_0(C^A(x))$ for almost all x. Choose an order b such that $b(\alpha(n, n, q)) \leq nh(|q|)$ for all n, q.

Let q_x be a minimal A-program for x, that is, $U^A(q_x) = x$ and $|q_x| = C^A(x)$. Let $n_x = C(x|q_x)$. Then $\Psi^A(m, n_x, q_x) \downarrow$ iff the m-th bit of p_x is 1, where p_x is the first program such that $|p_x| = n_x$ and $\tilde{U}(p_x, q_x) = x$.

Since A' is ω-r.e. via b, $p_x = A'(\alpha(1, n_x, q_x)) \ldots A'(\alpha(n_x, n_x, q_x))$ changes at most

$$n_x \max\{b(\alpha(m, n_x, q_x)) \colon 1 \leq m \leq n_x\} \leq n_x b(\alpha(n_x, n_x, q_x)) \leq n_x^2 h(|q_x|)$$

many times. Since $\tilde{U}(p_x, q_x) = x$ and we can describe p_x with n_x, q_x and the number of changes of $A'(\alpha(1, n_x, q_x)) \ldots A'(\alpha(n_x, n_x, q_x))$, we have

$$(1) \qquad n_x = C(x|q_x) \leq 2|n_x| + |n_x^2 h(|q_x|)| + \mathcal{O}(1) \leq 4|n_x| + |h(|q_x|)| + \mathcal{O}(1).$$

To finish, let us prove that for almost all x, $n_x \leq 2|h(|q_x|)| + \mathcal{O}(1)$. Since $C(x) \leq |q_x| + 2n_x + \mathcal{O}(1)$, this upper bound of n_x will imply that

$$C(x) \leq |q_x| + h(|q_x|) + \mathcal{O}(1) = \hat{h}(C^A(x)) + \mathcal{O}(1),$$

for almost all x, as we wanted. Hence, let us see that $n_x \leq 2|h(|q_x|)| + \mathcal{O}(1)$ for almost all x. There is a constant N such that for all $n \geq N$, $8|n| \leq n$. We know that for almost all x, q_x satisfies $|h(|q_x|)| \geq N$. Suppose x has this property. Then either $n_x \leq |h(|q_x|)|$ or $4|n_x| \leq n_x/2$. In the second case $n_x - 4|n_x| \geq n_x/2$ and by (1), $n_x/2 \leq |h(|q_x|)| + \mathcal{O}(1)$. So, in both cases, we have $n_x \leq 2|h(|q_x|)| + \mathcal{O}(1)$. □

Lemma 5.2 *For all $x \in \{0,1\}^*$ and $d \in \mathbb{N}$,*

$$|\{y : C(x,y) \leq C(x) + d\}| \leq \mathcal{O}(d^4 2^d).$$

Theorem 5.3 *The following are equivalent:*

(i) *A is strongly jump-traceable;*

(ii) *For every order h and almost every x, $C(x) \leq C^A(x) + h(C^A(x))$.*

Proof. (ii)⇒(i). Since there are at most $2^n - 1$ programs of length $< n$, $\forall n\ \exists x\ [|x| = n\ \wedge\ n \leq C(x)]$. Let c such that $\forall x\ C^A(x, J^A(|x|)) \leq |x| + c$. This last inequality holds because, given x, we can compute $J^A(|x|)$ relative to A.

For any function f, let $\hat{f}(y) = y + f(y)$ for all y. Let h be any order and let us prove that A is jump-traceable via h. Define the order g such that for almost all e, $3^{g(e+c)} \leq h(e)$. By hypothesis, for almost all x, if $J^A(x)\downarrow$ then $C(x, J^A(|x|)) \leq \hat{g}(C^A(x, J^A(|x|))) \leq |x| + g(|x| + c) + c$.

Define the trace $T_e = \{y : \forall x\ [|x| = e\ \Rightarrow\ C(x,y) \leq e + g(e+c) + c]\}$. It is clear that for almost all e, if $J^A(e)\downarrow$ then $J^A(e) \in T_e$, because given x such that $|x| = e$, we have $C(x, J^A(e)) \leq e + g(e+c) + c$. To verify that for almost all e, $|T_e| \leq h(e)$, suppose $y \in T_e$. Take x, $|x| = e$ and $C(x) \geq e$. Then

$$C(x,y) \leq e + g(e+c) + c \leq C(x) + g(e+c) + c.$$

By Lemma 5.2, for almost all e there are at most $3^{g(e+c)} \leq h(e)$ such y's in T_e.

(i)⇒(ii). Let h_0 be a given order. As in the proof of Theorem 5.1, it is sufficient to show that $C(x) \leq \hat{h}(C^A(x)) + \mathcal{O}(1)$ for almost all x, where $h = \lfloor h_0/2 \rfloor$. Take α and T as in Proposition 6.2 (part ii) with bound g such that $g(\alpha(x)) \leq h(|str(x)|)$. Let $m \in \mathbb{N}$ be such that $U^A(str(m)) = y$ and $|str(m)| = C^A(y)$. Since $y \in T_{\alpha(m)}$, we can code y with m and a number not greater than $g(\alpha(m))$ (representing the time in which y is enumerated into $T_{\alpha(m)}$), using at most $|str(m)| + g(\alpha(m)) \leq C^A(y) + h(C^A(y))$ bits. Then $\forall y\ C(y) \leq \hat{h}(C^A(y)) + \mathcal{O}(1)$. □

In [9], it was proven that there is a super-low which is not jump-traceable (namely, a super-low Martin-Löf random set). In contrast, from Theorem 5.1

and Theorem 5.3 we can conclude that the strong version of super-lowness implies strong jump-traceability.

Corollary 5.4 *If A' is well-approximable then A is strongly jump-traceable.*

6 Variations on K-triviality

Throughout this section, let $p : \mathbb{N} \to \mathbb{N}$ be nondecreasing such that $\lim_n p(n) - n = \infty$. Recall that A is K-trivial iff $\exists c\ \forall n\ K(A \restriction n) \leq K(n) + c$. Nies [8] showed that A is K-trivial if and only if A is low for K, i.e. $\exists c\ \forall x\ K(x) \leq K^A(x) + c$. In this section we weaken the notion of lowness for K:

Definition 6.1 A set A is *p-low* iff $\forall y\ K(y) \leq p(K^A(y) + c_0) + c_1$ for some constants c_0 and c_1. Let $\mathcal{M}[p]$ denote the class of such sets.

Clearly, if A is K-trivial then A is p-low and for every p (which we consider in this section). If $A \in \mathcal{M}[p]$ and $B \leq_T A$, then $B \in \mathcal{M}[p]$. Indeed, since $B \leq_T A$, there exists a constant c_2 such that for each string y, $K^A(y) \leq K^B(y) + c_2$. Then $K(y) \leq p(K^A(y) + c_0) + c_1 \leq p(K^B(y) + c_0 + c_2) + c_1$.

The following proposition states a relation between jump-traceability and p-lowness. In Theorem 5.3 we proved a similar result, involving strong jump-traceability and plain Kolmogorov complexity.

Proposition 6.2 (i) *Suppose p is a recursive function. There is a constant c such that if $A \in \mathcal{M}[p]$ via constants c_0 and c_1 then A is jump-traceable via $h(x) = 2^{p(2|x|+c_0+c)+c_1+1}$;*

(ii) *There is a reduction function α such that if A is jump-traceable via h then $A \in \mathcal{M}[p]$ for $p(z) = 3z + 2|h(\alpha(2^{z+1}))|$.*

Proof. For (i), we know that there is a constant c such that $K^A(J^A(x)) \leq 2|x| + c$ because we can compute $J^A(x)$ from x and the oracle A. Define the trace $T_x = \{U(\sigma) : |\sigma| \leq p(2|x| + c_0 + c) + c_1\}$. Clearly $|T_x| \leq 2^{p(2|x|+c_0+c)+c_1+1}$. Let $y = J^A(x)$. By hypothesis $K(y) \leq p(K^A(y) + c_0) + c_1$ and then $K(y) \leq p(2|x| + c + c_0) + c_1$. Hence $y \in T_x$.

For (ii), let α be a reduction function such that $J^A(\alpha(x)) = U^A(str(x))$. Let T be a trace for J^A with bound h and let us define the trace $\tilde{T}_n = \bigcup_{x:|str(x)|=n} T_{\alpha(x)}$. Notice that $|\tilde{T}_n| \leq \sum_{x:|str(x)|=n} h(\alpha(x)) \leq 2^n h(\alpha(2^{n+1}))$, since α is increasing. Let $m \in \mathbb{N}$ be such that $U^A(str(m)) = y$ and $|str(m)| = K^A(y)$. Since $y \in T_{\alpha(m)}$, we know that $y \in \tilde{T}_{|str(m)|}$, hence we describe y by saying "y is the i-th element enumerated into $\tilde{T}_{|str(m)|}$". If we code $|str(m)|$ in unary and we code i with $2|i| \leq 2|2^{|str(m)|} h(\alpha(2^{|str(m)|+1}))| \leq 2|str(m)| + 2|h(\alpha(2^{|str(m)|+1}))|$ many bits, we have $K(y) \leq p(K^A(y)) + \mathcal{O}(1)$, for $p(z) = 3z + 2|h(\alpha(2^{z+1}))|$. □

Corollary 6.3 *A is jump-traceable iff there exists a recursive function p (of the type considered in this section) such that* $A \in \mathcal{M}[p]$.

Acknowledgement

The authors acknowledge support from the Marsden fund, Royal Society of New Zealand.

References

[1] K. Ambos-Spies, C. Jockusch and R. Shore, *An algebraic decomposition of the recursively enumerable degrees and classes equal to the promptly simple degrees*, Transactions of the American Mathematical Society 281:109–128, 1984.

[2] M. Bickford and F. Mills, *Lowness properties of r.e. sets*, Manuscript, UW Madison, 1982.

[3] C. Calude, P. Hertling, B. Khoussainov and Y. Wang, *Recursively enumerable reals and Chaitin's* Ω *number*, in STACS 1998, Lecture Notes in Computer Science 1373:596–606, 1998.

[4] G. Chaitin, *A theory of program size formally identical to information theory*, Journal of the Association for Computing Machinery 22:329–340, 1975.

[5] R. Downey, D. Hirschfeldt, A. Nies and F. Stephan, *Trivial reals*, Proceedings of the 7th and 8th Asian Logic Conferences, World Scientific, River Edge, NJ, pages 103–131, 2003.

[6] B. Kjos-Hanssen, W. Merkle and F. Stephan, *Kolmogorov complexity and the Recursion theorem*, Manuscript, 2005.

[7] J. Mohrherr, *A refinement of low n and high n for the r.e. degrees*, Z. Math. Logik Grundlag. Math., 32(1):5-12, 1986.

[8] A. Nies, *Lowness properties and randomness*, to appear in Advances in Math.

[9] A. Nies, *Reals which compute little*, CDMTCS Research Report 202, Dec. 2002.

[10] P.G. Odifreddi, "Classical recursion theory" Volume 1, North-Holland, Amsterdam 1989, Volume 2, Elsevier, Amsterdam 1999.

[11] R. Soare, *Recursively enumerable sets and degrees*, Springer, Heidelberg, 1987.

[12] S. Terwijn, D. Zambella, *Algorithmic randomness and lowness*, J. Symbolic Logic, 66:1199-1205, 2001.