Available online at www.sciencedirect.com
ScienceDirect
Annals of Pure and Applied Logic 152 (2008) 51–66
ANNALS OF PURE AND APPLIED LOGIC
www.elsevier.com/locate/apal

Lowness properties and approximations of the jump

Santiago Figueira[a,*], André Nies[b], Frank Stephan[c]

[a] Departamento de Computación, FCEyN, Universidad de Buenos Aires, Pabellón I, Ciudad Universitaria (C1428EGA), Buenos Aires, Argentina
[b] Department of Computer Science, University of Auckland, Building 303, 38 Princes Street, Auckland, New Zealand
[c] School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, 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 super-lowness for sets of natural numbers. A computable non-decreasing unbounded function h is called an order function. Informally, a set A is strongly jump-traceable if for each order function h, for each input e one may effectively enumerate a set T_e of possible values for the jump $J^A(e)$, and the number of values enumerated is at most $h(e)$. A' is well-approximable if can be effectively approximated with less than $h(x)$ changes at input x, for each order function h. 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 strong jump-traceability in terms of Kolmogorov complexity. We also investigate other properties of these lowness properties.

MSC: 68Q30; 03D25; 03D35

Keywords: Lowness; Traceability; ω-r.e.; K-triviality; Kolmogorov complexity

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 [10], Nies proved that a set is K-trivial if and only if A is low for Martin-Löf-random (that is, each Martin-Löf-random set is already random relative to A).

Terwijn and Zambella [13] 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 $f(x) \in D_{r(x)}$. They showed that this combinatorial notion characterizes the sets that are low for Schnorr tests.

* Corresponding author.
E-mail addresses: sfigueir@dc.uba.ar (S. Figueira), nies@cs.auckland.ac.nz (A. Nies), fstephan@comp.nus.edu.sg (F. Stephan).

0168-0072/$ - see front matter
doi:10.1016/j.apal.2007.11.002

This property was modified in [11] 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. A uniform 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 [11] is *super-lowness*, first introduced in [1,9].

Definition 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, that is, $|\{s : g(x, s) \neq g(x, s+1)\}| \leq b(x)$. 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 an ω-r.e. set.

Recall that a set A is *low* if $A' \leq_T \emptyset'$. The above definition of A being super-low is equivalent to $A' \leq_{tt} \emptyset'$. Hence super-lowness implies lowness.

Both the classes of jump-traceable and of super-low sets are closed downward under Turing reducibility and contained in the class of generalized low sets $\{A : A' \leq A \oplus \emptyset'\}$. In [11] it was proven 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* and *well-approximability of the jump*, strengthening super-lowness. In the strong variant of these notions we consider *all* order functions as the bound instead of just *some* recursive bound. Here, an *order* function is a recursive, non-decreasing and unbounded function (intuitively, think of a *slowly growing* but unbounded recursive function). Our first two 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 an r.e. set.

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 Kolmogorov complexity relative to A is only a bit smaller than the unrelativized one. We prove some characterizations of jump-traceability and strong jump-traceability in terms of prefix-free (denoted by K) and plain (denoted by 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 function h and almost all x.

Recall that A is low for K iff $K(x) \leq K^A(x) + O(1)$ for each x. Nies [10] has shown that this property is equivalent to being K-trivial. In particular, non-computable low for K sets exist. The corresponding property involving C is only satisfied by the computable sets (because it implies being C-trivial by [3], which is the same as computable). The characterization of strongly jump-traceable is via a property that states that C^A is very close to C, while not implying computability.

By [10], K-triviality implies jump-traceability. Recently, Cholak, Downey and Greenberg [4] have shown that for r.e. sets A, strong jump-traceability implies K-triviality. They also prove that there is a K-trivial r.e. set that is not strongly jump-traceable.

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 by $A \upharpoonright n$ the string of length n which consists of the bits $A(0) \ldots A(n-1)$.

If A is given by an effective approximation and Ψ is a functional, we write $\Psi^A(e)[s]$ for $\Psi_s^{A_s}(e)$. 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 set 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

$$U^A(0^d 1\sigma) = M_d^A(\sigma)$$

and the prefix-free Kolmogorov complexity relative to A is defined as

$$K^A(y) = \min\{|\sigma| : U^A(\sigma) = y\},$$

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. Schnorr's Theorem states that $A \in \{0,1\}^\infty$ is Martin-Löf random iff the initial segments of A have high K-complexity, that is,

$$\exists c\ \forall n\ K(A \upharpoonright n) > n - c.$$

A set A is K-trivial iff the initial segments of A have low K-complexity, that is,

$$\exists c\ \forall n\ K(A \upharpoonright n) \le K(n) + c.$$

We say that $A \le_K B$ iff

$$\exists c\ \forall n\ K(A \upharpoonright n) \le K(B \upharpoonright n) + c.$$

The Kraft–Chaitin Theorem states that from a recursive sequence of pairs $(\langle n_i, \sigma_i\rangle)_{i\in\mathbb{N}}$ (known as *requests*) 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 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 given by

$$\tilde{U}(0^d 1\sigma, x) = \tilde{M}_d(\sigma, x),$$

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, that is,

$$C(y|x) = \min\{|\sigma| : \tilde{U}(\sigma, x) = y\}.$$

Let $str : \mathbb{N} \to \{0,1\}^*$ be the standard enumeration of the strings. The string $str(n)$ is that binary sequence $b_0 b_1 \ldots b_m$ for which the binary number $1b_0b_1\ldots b_m$ 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,

$$|W_e| = \infty \;\Rightarrow\; \exists s\ \exists x\ [x \in W_{e,s+1} \setminus W_{e,s}\ \wedge\ x \in A_{p(s)}]. \tag{1}$$

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 single maximal order function that suffices as the bounding function for all instances of jump-traceability.

Definition 3. A computable function $h : \mathbb{N} \to \mathbb{N}^+$ is an *order function* if h is non-decreasing and unbounded.

Notice that any reduction function is an order function.

Definition 4. A set A is *strongly jump-traceable* iff for each order function h, A is jump-traceable via h.

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

Proposition 5. $\{A : A$ *is strongly jump-traceable*$\}$ *is closed downward under Turing reducibility.*

Proof. Suppose that A is strongly jump-traceable, $B \leq_T A$. We prove that B is jump-traceable via the given order function h. Let Ψ be the functional such that $\Psi^A(x) = J^B(x)$ for all x and let α be the reduction function such that $J^A(\alpha(x)) = \Psi^A(x)$. We know that A is jump-traceable via a trace $(T_i)_{i\in\mathbb{N}}$ with bound $\tilde{h}$, where $\tilde{h}(z) = h(\min\{y : y \in \mathbb{N} \wedge \alpha(y+1) \geq z\})$. Observe that, since α is an order function, $\tilde{h}$ also is. Clearly,

$$J^B(e) = J^A(\alpha(e)) \downarrow \Rightarrow \ J^B(e) \in T_{\alpha(e)}.$$

Now, $\tilde{h}(\alpha(e)) = h(y)$ for some y such that $\alpha(y) < \alpha(e)$ or $y = 0$. Then $y \leq e$ and $\tilde{h}(\alpha(e)) = h(y) \leq h(e)$. Hence $(T_{\alpha(i)})_{i\in\mathbb{N}}$ is a trace for the jump of B with bound h. $\square$

Clearly each computable set A is strongly jump-traceable, because we can trace the jump by

$$T_e = \begin{cases} \{J^A(e)\} & \text{if } J^A(e)\downarrow; \\ \emptyset & \text{otherwise.} \end{cases}$$

In Theorem 7 below we show the existence of a non-computable strongly jump-traceable set. We need the following result, proven in [8, Theorem 2.3.1]:

Lemma 6. *The function* $m(x) = \min\{C(y) : y \geq x\}$ *is unbounded, non-decreasing and for every order function* f *there is an* x_0 *such that* $m(x) < f(x)$ *for all* $x \geq x_0$. *Also,* $m(x) = \lim_{s\to\infty} m_s(x)$, *where* $m_s(x) = \min\{C_s(y) : x \leq y \leq x+s\}$ *is recursive and* $m_s(x) \geq m_{s+1}(x)$, *for all* x *and* s.

Observe that here $\lambda x, s.C_s(x)$ is the standard recursive approximation from above of $C(x)$ (that is $\lambda s.C_s(x) \to C(x)$ when $s \to \infty$ and $C_s(x) \geq C_{s+1}(x)$).

Theorem 7. *There exist a promptly simple strongly jump-traceable set.*

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

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

These requirements will ensure that A is promptly simple (indeed, take $p(s) = s+1$ in Eq. (1)). Each time we enumerate an element into A in order to satisfy P_e, we may destroy $J^A(k)$ and then our trace for the jump of A will grow. Hence, we must enumerate elements into A in a controlled way, and sometimes we should refrain from putting elements into A. Since for any order function h there has to be a trace for J^A bounded by h, we will work with the function m defined in Lemma 6, which grows slower than any order function. The rule will be that during the construction, P_e may destroy $J^A(k)$ at stage s only if $e < m_s(k)$. (Observe that the restriction on P_e imposed rule may strengthen as s grows, because we may have $m_s(k) > m_{s+1}(k)$.) In this way, we will guarantee that the size of our trace for $J^A(e)$ will be bounded by $m(e)$, which will suffice because $m \leq h$ from some point on. As we will see, the exact choice of the trace for J^A with bound h depends on h, and is made in a non-uniform way.

In the following construction we use the convention that $W_{e,s} \subseteq \{0, 1, \ldots, s\}$ for all indices e and stages s.

Construction of A. Let m_s be the non-decreasing, unbounded function defined in Lemma 6.
Stage 0: set $A_0 = \emptyset$ and declare P_e unsatisfied for all e.
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 $m_s(k) \leq e$, if $J^A(k)[s]$ is defined then x is greater than the use of $J^A(k)[s]$.

If such e exists, put the least such x into A for each such e. 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 function h. We will prove that there exists an r.e. trace T for J^A as in Definition 1. Let h be any order function. By Lemma 6, there exists k_0 such that for all $k \geq k_0$, $m(k) \leq h(k)$. Define the recursive function

$$f(k) = \begin{cases} \min\{s : m_s(k) \leq h(k)\} & \text{if } k \geq k_0; \\ 0 & \text{otherwise.} \end{cases}$$

For $k \geq k_0$ and $s \geq f(k)$, $m_s(k)$ will be below $h(k)$, so $J^A(k)$ may change because P_e receives attention, for $e < m_s(k) \leq h(k)$. Since each P_e receives attention at most once, $J^A(k)$ can change at most $h(k)$ times after stage $f(k)$. So

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

is as required.

Fix e such that W_e is infinite and let us see that P_e is met. Let s such that

$$\forall k\ [m(k) \leq e \Rightarrow m_s(k) = m(k)]$$

and $s' > s$ such that no P_i receives attention after stage s' for any $i < e$. Then, by the construction, no computation $J^A(k)$, $m(k) \leq e$ can be destroyed after stage s'. So there is $t > s'$ such that for all k where $m_t(k) \leq e$, 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 $m_{t'}(k) \leq e$. Now either P_e was already satisfied or P_e receives attention at stage $t' + 1$. In either case P_e is met. $\square$

Next we study the size of the trace bound for jump-traceability. Given an order function h, it is 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.

Theorem 8. *For any order function h there is an r.e. set A and an order function $\tilde{h}$ 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 (that is, $\Psi^X(e) = J^X(\alpha(e))$ 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 function $\tilde{h}$ 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 eth such trace, is as in Definition 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, that is,

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

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 in the construction 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.

Construction of A. We define $\tilde{T} = \{\tilde{T}_0, \tilde{T}_1, \dots\}$ by stages. The sth 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 $j \leq s$ and P_j for $j < s$. We also initialize the new P_s. We look at the least procedure requiring attention in the order

$$P_0, N_0, \dots, P_s, N_s.$$

If there is none, do nothing. Otherwise, suppose that 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 $e < j \leq s$. In this case, we say that N_e *acts*.

Verification. Let us prove that P_e is met. Take s such that all $J^A(i)$ are stable for $i < e$. Suppose that 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 less or equal than 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 ith 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 function and it constitutes a bound for the trace $(\tilde{T}_i)_{i\in\mathbb{N}}$. $\square$

It is open whether there is *minimal* bound for jump-traceability. That is, given an order function h, there is a set A and an order function $\tilde{h}$ such that A is jump-traceable via h but not via a smaller function $\tilde{h}$? If the answer is negative for some order function h, then strong jump-traceability is equivalent jump-traceability for that single function h.

4. Well-approximability of the jump

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

Definition 9. A set D is *well-approximable* iff for each order function b, D is ω-r.e. via b.

Clearly, if A' is well-approximable, then A is super-low. It is not difficult to see that well-approximability of the jump is closed downward under Turing reducibility.

Proposition 10. $\{A : A'$ *is well approximable*$\}$ *is closed downward under Turing reducibility.*

Proof. Suppose A is such that A' is well-approximable and let $B \leq_T A$. We prove that B' is well-approximable via the given order function b. Define Ψ and α as in Proposition 5. We know that there is a recursive $\{0, 1\}$-valued g such that $A'(x) = \lim_{s\to\infty} g(x, s)$ and $g(x, s)$ changes at most $\tilde{b}(x)$ times, where $\tilde{b}(z) = b(\min\{y : y \in \mathbb{N} \wedge \alpha(y+1) \geq z\})$. Then

$$\lim_{s\to\infty} g(\alpha(x), s) = A'(\alpha(x)) = B'(x)$$

and $g(\alpha(x), s)$ changes at most $\tilde{b}(\alpha(x))$ times. As in Proposition 5, $\tilde{b}(\alpha(x)) \leq b(x)$. $\square$

We next prove that if A is r.e., then A is strongly jump-traceable iff A' is well-approximable. We will need the following two lemmas.

Lemma 11. *Let f and $\hat{f}$ be order functions 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 $\hat{f}$.*

Proof. Assume that $\exists x_0\ \forall x\ [x \geq x_0 \Rightarrow f(x) \leq \hat{f}(x)]$. For (i), suppose that T is a trace for J^A with bound f. We can define the trace $\hat{T}$:

$$\hat{T}_x = \begin{cases} T_x & \text{if } x \geq x_0; \\ \{J^A(x)\} & \text{otherwise.} \end{cases}$$

Hence, if $x \geq x_0$ then $|\hat{T}_x| = |T_x| \leq f(x) \leq \hat{f}(x)$, and if $x < x_0$ then $1 = |\hat{T}_x| \leq \hat{f}(x)$.

For (ii), suppose that A is well-approximable via the $\{0, 1\}$-valued $g(x, s)$ which changes at most $f(x)$ times. Define

$$\hat{g}(x, s) = \begin{cases} g(x, s) & \text{if } x \geq x_0; \\ A(x) & \text{otherwise.} \end{cases}$$

If $x \geq x_0$ then $\hat{g}(x, s)$ changes at most $f(x) \leq \hat{f}(x)$ times, and if $x < x_0$ then $\hat{g}$ does not change at all. $\square$

Lemma 12. *There exists a recursive γ such that for all r.e. A:*

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

Proof. We follow the proof of [11, Theorem 4.1], together with Lemma 11.

(i) $\Rightarrow$(ii). Suppose that A is jump-traceable via h. By [11] A is super-low via a $\{0, 1\}$-valued recursive g such that $g(x, s)$ changes at most $2h(\alpha(x)) + 2$ times. Here, α is a reduction function (hence primitive recursive) which depends on A. The diagonal γ of the Ackermann-function satisfies $\gamma(x) \geq \alpha(x)$ for almost all x [12, Volume 2, Theorem VIII.8.10]. Since h is an order function, $2(h \circ \gamma) + 2$ also is, and $2h(\gamma(x)) + 2 \geq 2h(\alpha(x)) + 2$ for almost all x. By Lemma 11, A is super-low via $b(x) = 2h(\gamma(x)) + 2$.

(ii) $\Rightarrow$(i). Suppose that A is super-low via an order function b and the $\{0, 1\}$-valued function g. Again following [11], there is a trace for J^A via $\lfloor \frac{1}{2}(b \circ \gamma) \rfloor$, for a primitive recursive α which depends on g. As we did in the previous implication, $\lfloor \frac{1}{2} b(\gamma(x)) \rfloor \geq \lfloor \frac{1}{2} b(\alpha(x)) \rfloor$ for almost all x. Thus A is jump-traceable via $h(x) = \lfloor \frac{1}{2} b(\gamma(x)) \rfloor$. $\square$

Theorem 13. *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) $\Rightarrow$(ii). Given an order function b, let us prove that A is super-low via b. By part (i) of Lemma 12, it suffices to define an order function 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 function.

(ii) $\Rightarrow$(i). Given an order function h, we will prove that A is jump-traceable via h. By part (ii) of Lemma 12, it suffices to define an order function 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. $\square$

Later, in Corollary 18, we will improve this result and we will see that, in fact, the implication (ii) $\Rightarrow$(i) holds for any A.

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

Theorem 14. *If D is well-approximable then for almost all n, $K(D \upharpoonright n) \leq 4|n|$.*

Proof. Suppose that $D(n) = \lim_{s\to\infty} g(n, s)$, where g is recursive and changes at most n times. Given n, there is a unique s and some $m < n$ such that $g(m, s) \neq g(m, s + 1)$ but $g(q, t) = g(q, t + 1)$ for all $t > s$ and $q < n$. That is, s is the first stage where $g(0, s + 1) = D(0), \dots, g(n - 1, s + 1) = D(n - 1)$ 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 $g(m, t) \neq g(m, t + 1)$. So one can compute $D \upharpoonright n$ 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 additional 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

$$\forall n\ K(D \upharpoonright n) \leq 3|n| + |n|/2 + d$$

and then the relation $K(D \upharpoonright n) \leq 4|n|$ holds for almost all n. In fact, using binary notation to store q instead of 1^q0, it would even give

$$K(D \upharpoonright n) \leq 3(|n| + \log(|n|))$$

for almost all n. $\square$

5. Traceability and plain Kolmogorov complexity

We give a characterization of strong jump-traceability in terms of (relativized) plain Kolmogorov complexity. First we show that if A' is well-approximable then A satisfies the condition involving Kolmogorov complexity and hence that any set A such that A' is well-approximable is strongly jump-traceable.

Theorem 15. *If A' is well-approximable then for every order function h and almost all x, $C(x) \leq C^A(x)+h(C^A(x))$.*

Proof. The idea of the proof is the following. Let h be any order function. Suppose that q_x is a minimal A-program for x. We know that there is a c such that $C(x) \leq |q_x| + 2C(x|q_x) + c$. Since $|q_x| = C^A(x)$, we only need to show that $2C(x|q_x) + c \leq h(|q_x|)$ for almost all x. Given q_x and the value of $C(x|q_x)$, we can find a program p_x of length $C(x|q_x)$ which describes x with the help of q_x, that is $\tilde{U}(p_x, q_x) = x$. It can be shown that there is a recursive $\{0, 1\}$-valued approximation of the bits of p_x which changes few times (in the proof, this is done with the help of the functional Ψ). Hence, x can be described by the values of $C(x|q_x)$, q_x and p_x. We can represent p_x with the number of changes of the mentioned $\{0, 1\}$-valued approximation. This will show that $C(x|q_x) \leq 2|h(|q_x|)| + O(1)$, which is sufficient to get the desired upper bound on $2C(x|q_x) + c$.

Here are the details. 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 mth 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 $J^A(\alpha(m, n, q)) = \Psi^A(m, n, q)$. Choose an order function b such that $b(\alpha(n, n, q)) \leq nh(|q|)$ for all n, q. We can approximate $A'(x)$ with a $\{0, 1\}$-valued recursive function which changes at most $b(x)$ times.

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 mth 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,

$$p_x = A'(\alpha(1, n_x, q_x)) \dots A'(\alpha(n_x, n_x, q_x))$$

changes at most

$$\begin{aligned} n_x \max\{b(\alpha(m, n_x, q_x)) : 1 \leq m \leq n_x\} &\leq n_x b(\alpha(n_x, n_x, q_x)) \\ &\leq n_x^2 h(|q_x|) \end{aligned}$$

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

$$\begin{aligned} n_x = C(x \mid q_x) &\leq 2|n_x| + |n_x^2 h(|q_x|)| + \mathrm{O}(1) \\ &\leq 4|n_x| + |h(|q_x|)| + \mathrm{O}(1). \end{aligned} \tag{2}$$

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

$$\begin{aligned} C(x) &\leq |q_x| + h(|q_x|) \\ &= C^A(x) + h(C^A(x)) \end{aligned}$$

for almost all x, as we wanted. Hence, let us see that $n_x \leq 2|h(|q_x|)| + \mathrm{O}(1)$ 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 that 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 (2), $n_x/2 \leq |h(|q_x|)| + \mathrm{O}(1)$. So, in both cases, we have $n_x \leq 2|h(|q_x|)| + \mathrm{O}(1)$. □

To characterize strong jump-traceability, we need a lemma.

Lemma 16. *For all $x \in \{0, 1\}^*$ and $d \in \mathbb{N}$,*

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

Proof. Chaitin [2] has proved that

$$\forall d, n \in \mathbb{N}\ |\{\sigma : |\sigma| = n \ \wedge\ C(\sigma) \leq C(n) + d\}| = \mathrm{O}(2^d).$$

Let c be such that $\forall x\ C(x) \leq str^{-1}(x) + c$. Consider the partial recursive function $f(x, y, d)$ which enumerates all strings z such that $C(z) \leq str^{-1}(x) + d + c$ until it finds $z = y$. If z was the ith string to appear in the enumeration, then $f(x, y, d)$ is the number i written in binary with initial zeroes such that $|f(x, y, d)| = str^{-1}(x) + d + c + 1$. Notice that it is always possible to write $f(x, y, d)$ in this way because there are at most $2^{str^{-1}(x)+d+c+1}$ such strings z. If no such z exists, then $f(x, y, d) \uparrow$. Let x and d be given. Consider y such that $C(x, y) \leq C(x) + d$. Since $C(x, y) \leq str^{-1}(x) + d + c$ then $f(x, y, d) \downarrow$ and

$$\begin{aligned} C(f(x, y, d)) &\leq C(x, y) + 2|d| + \mathrm{O}(1) \\ &\leq C(x) + d + 2|d| + \mathrm{O}(1) \\ &\leq C(str^{-1}(x) + d + c + 1) + d + 4|d| + \mathrm{O}(1). \end{aligned}$$

The last inequality holds because we can compute the string x from the numbers $str^{-1}(x) + d + c + 1$ and d. Let $n = str^{-1}(x) + d + c + 1$ and $d' = d + 4|d| + \mathrm{O}(1)$. For fixed x and d, the mapping $y \mapsto f(x, y, d)$ is injective and thus

$$\begin{aligned} |\{y : C(x, y) \leq C(x) + d\}| &\leq |\{\sigma : |\sigma| = n \ \wedge\ C(\sigma) \leq C(n) + d'\}| \\ &= \mathrm{O}(2^{d'}) = \mathrm{O}(d^4 2^d). \end{aligned}$$

This completes the proof. □

Theorem 17. *The following are equivalent:*

(i) *A is strongly jump-traceable;*
(ii) *For every order function h and almost every x, $C(x) \leq C^A(x) + h(C^A(x))$.*

Proof. For any function f, let $\hat{f}(y) = y + f(y)$ for all y.

(i) ⇒(ii). Let h_0 be a given order function. It is sufficient to show that $C(x) \leq \hat{h}(C^A(x)) + \mathrm{O}(1)$ for almost all x, where $h = \lfloor h_0/2 \rfloor$. 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 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 place ($\leq g(\alpha(m))$) within the enumeration of $T_{\alpha(m)}$ at which y is enumerated), using at most

$$|str(m)| + g(\alpha(m)) \leq C^A(y) + h(C^A(y))$$

many bits. Then $\forall y\ C(y) \leq \hat{h}(C^A(y)) + O(1)$.

(ii) $\Rightarrow$(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 be a constant such that

$$\forall x\ [J^A(|x|)\downarrow \Rightarrow\ C^A(x, J^A(|x|)) \leq |x| + c].$$

This last inequality holds because, given x, we can compute $J^A(|x|)$ relative to A.

Let h be any order function and let us prove that A is jump-traceable via h. Define the order function 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

$$\begin{aligned} C(x, J^A(|x|)) &\leq \hat{g}(C^A(x, J^A(|x|))) \\ &\leq |x| + g(|x| + c) + c. \end{aligned}$$

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 that $y \in T_e$. Take x, $|x| = e$ and $C(x) \geq e$. Then

$$\begin{aligned} C(x, y) &\leq e + g(e + c) + c \\ &\leq C(x) + g(e + c) + c. \end{aligned}$$

By Lemma 16, for almost all e there are at most $3^{g(e+c)} \leq h(e)$ such y's in T_e. $\square$

In [11], it was proven that there is a super-low set which is not jump-traceable (namely, a super-low Martin-Löf random set). In contrast, from Theorems 15 and 17 we can conclude that the strong version of super-lowness implies strong jump-traceability.

Corollary 18. *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 strictly increasing such that in addition $\lim_n p(n) - n = \infty$. We call p an *estimation function* if, in addition, $p(n) = \lim_s p_s(n)$ where $p_{s+1}(n) \leq p_s(n)$, and the function $\lambda s, n.p_s(n)$ is recursive. An example of such a function is $q(n) = n + 5 \cdot \min\{K(m) : m \geq n\}$ with the approximation $q_s(n) = n + 5 \cdot \min\{K_s(m) : s \geq m \geq n\}$.

Recall that A is K-trivial iff

$$\exists c\ \forall n\ K(A \upharpoonright n) \leq K(n) + c.$$

Nies [10] has shown that A is K-trivial if and only if A is low for K, that is, $\exists c\ \forall x\ K(x) \leq K^A(x) + c$. In this section we weaken the notion of lowness for K:

Definition 19. (i) A set A is *weakly p-low* iff $\forall n\ K(A \upharpoonright n) \leq p(K(n) + c_0) + c_1$ for some constants c_0 and c_1. Let $\mathcal{K}[p]$ denote the class of such sets.

(ii) 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.

Proposition 20. (i) *If $A \in \mathcal{M}[p]$ and $B \leq_T A$, then $B \in \mathcal{M}[p]$.*

(ii) *If $A \in \mathcal{K}[p]$ and either $B \leq_K A$ or $B \leq_{wtt} A$, then $B \in \mathcal{K}[p]$.*

(iii) *Suppose that p is an estimation function. Then no random set is in $\mathcal{K}[p]$.*

(iv) *If* $A, B \in \mathcal{K}[p]$ *and* A, B *are r.e., then*

$$A \oplus B = \{2x : x \in A\} \cup \{2x + 1 : x \in B\} \in \mathcal{K}[p].$$

(v) $\mathcal{M}[p] \subseteq \mathcal{K}[p]$.

Proof. (i) 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

$$\begin{aligned} K(y) &\leq p(K^A(y) + c_0) + c_1 \\ &\leq p(K^B(y) + c_0 + c_2) + c_1. \end{aligned}$$

(ii) This is trivial for $\leq_K$. Now suppose that $B = \Gamma^A$ for a weak truth-table reduction Γ with recursive bound f. Without loss of generality, we may assume that f strictly increasing. Given $A \upharpoonright f(n)$ we can compute n and $B \upharpoonright n$, and then there is a constant c_2 such that for all n,

$$\begin{aligned} K(B \upharpoonright n) &\leq K(A \upharpoonright f(n)) + c_2 \\ &\leq p(K(f(n)) + c_0) + c_1 + c_2. \end{aligned}$$

Since f is recursive, we have $K(f(n)) \leq K(n) + \mathrm{O}(1)$ and hence $B \in \mathcal{K}[p]$.

(iii) Assume that $\forall n\ K(A \upharpoonright n) > n - c$ and $A \in \mathcal{K}[p]$ via constants c_0 and c_1. Since p is an estimation function, $p(n) = \lim_s p_s(n)$ where $p_{s+1}(n) \leq p_s(n)$, and the function $\lambda s, n.p_s(n)$ is recursive. Define the strictly increasing recursive function $\tilde{p}(0) = p_0(0)$ and $\tilde{p}(k+1) = p_0(j)$, where $j = \min\{i : i > k \ \wedge\ p_0(i) > \tilde{p}(k)\}$. Since $\tilde{p} \geq p$, $A \in \mathcal{K}[\tilde{p}]$. Define the Kraft–Chaitin set $\{\langle i, n_i\rangle : i \in \mathbb{N}^+ \ \wedge\ n_i = \tilde{p}(i + d + c_0) + c_1 + c\}$ for M_d with d given in advance by the Recursion Theorem. Then $K(n_i) \leq i + d$ and hence $\tilde{p}(K(n_i) + c_0) \leq \tilde{p}(i + d + c_0)$. Finally,

$$\begin{aligned} K(A \upharpoonright n_i) &\leq \tilde{p}(K(n_i) + c_0) + c_1 \\ &\leq \tilde{p}(i + d + c_0) + c_1 = n_i - c \end{aligned}$$

and this is a contradiction.

(iv) Ignoring constants, for each n,

$$\begin{aligned} K(A \oplus B \upharpoonright n) &\leq K(A \oplus B \upharpoonright 2n) \\ &\leq \max\{K(A \upharpoonright n), K(B \upharpoonright n)\} \\ &\leq p(K(n)). \end{aligned}$$

In the second inequality we used [6, Theorem 6.4].

(v) Again ignoring constants, for all n,

$$\begin{aligned} K(A \upharpoonright n) &\leq p(K^A(A \upharpoonright n)) \\ &\leq p(K^A(n)) \\ &\leq p(K(n)). \end{aligned}$$

This completes the proof. □

The following proposition shows a connection between jump-traceability and p-lowness. In Theorem 17 we proved a similar result, relating strong jump-traceability and plain Kolmogorov complexity.

Proposition 21. (i) *Suppose that 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

$$\begin{aligned} |\tilde{T}_n| &\leq \sum_{x:|str(x)|=n} h(\alpha(x)) \\ &\leq 2^n h(\alpha(2^{n+1})), \end{aligned}$$

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 ith element enumerated into $\tilde{T}_{|str(m)|}$". If we code $|str(m)|$ in unary and we code i with

$$\begin{aligned} 2|i| &\leq 2|2^{|str(m)|} h(\alpha(2^{|str(m)|+1}))| \\ &\leq 2|str(m)| + 2|h(\alpha(2^{|str(m)|+1}))| \end{aligned}$$

many bits, we have $K(y) \leq p(K^A(y)) + O(1)$, for $p(z) = 3z + 2|h(\alpha(2^{z+1}))|$. □

Corollary 22. *A is jump-traceable iff there exists a recursive function p (of the type considered in this section) such that $A \in \mathcal{M}[p]$.*

Figueira, Stephan and Wu [7, Proposition 6] used a universal machine which has the property that there is an approximation K_s of K from the above with $K_x(x) = K(x)$ for all $x \in X$ where $X = \{x : \forall y > x\,(K(y) > K(x))\}$. For the following Theorem, such a universal machine is assumed. The example shows that there is a set in $\mathcal{M}[q]$ where q is as defined at the beginning of Section 6 which is not K-trivial. Note that r differs from the function in Lemma 6 only by using K instead of C and has the same properties as the function given there. In particular, for each order function h we have $r(n) \leq n + h(n)$ for almost each n, and thus the set constructed satisfies the analog for K of the condition in Theorem 17 characterizing strong jump traceability. In contrast to this result, Cholak, Downey and Greenberg [4] have shown that each strongly jump-traceable set is in Δ^0_2.

Theorem 23. *Let $r(n) = \min\{K(m) : m \geq n\}$ and $q(n) = n + 5 \cdot r(n)$. Then there is a set $A \in \mathcal{M}[q] \setminus \Delta^0_2$.*

Proof. Note that the set $X = \{x : \forall y > x\, \forall t\,(K_t(y) > K_x(x))\}$ is co-r.e. and that it has a co-r.e. subset Y of the form $\{y_0, y_1, \ldots\}$ such that, for all n, $y_n = K(y_{n+1}) = K_{y_{n+1}}(y_{n+1})$. As $K(0) > 0$ one might have the undesirable property that $y_{n+1} < y_n$ for some n. But as there are only finitely many numbers x with $K(x) > x$, one simply adds to the construction of Y the condition that y_0 is taken to be the first element of X larger than these finitely many exceptions and so one has the additional property that $y_{n+1} > y_n$ for all n.

Now one defines a partition $I_0, I_1, \ldots$ of the natural numbers into intervals such that $|I_x| = K_x(K_x(x))$ and $\max(I_x) + 1 = \min(I_{x+1})$. Note that none of these intervals is empty as $K_x(K_x(x)) > 0$ for all x which is due to the fact that a prefix-free universal machine is undefined on the empty input.

Having the partition, one defines a partial-recursive function ψ in stages s where one does the following algorithm where ψ is everywhere undefined before stage 0. The set A will be chosen such that its characteristic function is a suitable extension of ψ. Let ψ_s denote the approximation to ψ before stage s.

- Find the least x, y such that $x \leq s$, $y \in I_x$, $\psi_s(y)$ is undefined and either (1) $x \notin Y_s$ or (2) there is a string $\sigma \in \{0,1\}^{\max(I_x)+1}$ such that $K_s(\sigma) < K_s(x) + 0.5 \cdot \log(|I_x|)$ and σ is consistent with ψ_s, that is, $\psi_s(z) = \sigma(z)$ for all $z \in \text{domain}(\psi_s) \cap \{0, 1, \ldots \max(I_x)\}$.
- In the case that no x, y were found, let $\psi_{s+1} = \psi_s$.
- In the case that x, y were found according to condition (1), let $\psi_{s+1}(y) = 0$ and let $\psi_{s+1}(z) = \psi_s(z)$ for all $z \neq y$.
- In the case that x, y were found according to condition (2), let $\psi_{s+1}(y) = 1 - \sigma(y)$ and let $\psi_{s+1}(z) = \psi_s(z)$ for all $z \neq y$.

Now let A be a set whose characteristic function extends ψ and which is low for Ω. Such a set A exists since ψ defines a Π^0_1 class and Downey, Hirschfeldt, Miller and Nies [5] showed every Π^0_1 class (of sets) has a member which is low for Ω.

Reviewing the construction of ψ, condition (1) enforces that ψ is defined on the complete interval I_x if $x \notin Y$ and condition (2) enforces that if $x = y_n$ and n is large enough then the Kolmogorov complexity of $A \upharpoonright \max(I_{y_n})$ is at least $K(y_n) + \log(|I_{y_n}|)/2$. To see this, one should have in mind that $x \to \max(I_x)$ is a recursive injective function, that $K_{y_n}(y_n) = K(y_n)$ and that the number of σ of length $\max(I_{y_n}) + 1$ with $K(\sigma) \leq K(y_n) + \log(|I_{y_n}|)/2$ is bounded by a function proportional to $\sqrt{|I_{y_n}|}$. So there will for all sufficiently large n remain some elements in I_{y_n} where ψ is undefined. As the intervals I_{y_n} are of unbounded length, this enforces that for sufficiently large n the value of $K(A \upharpoonright \max(I_{y_n}))$ is at least $K(y_n) + \log(|I_{y_n}|)/2$ while on the other hand $K(\max(I_{y_n}))$ is only a constant above $K(y_n)$. So A is not K-trivial. Since every low for Ω set is either K-trivial or not Δ^0_2, A is also not Δ^0_2, that is, not limit-recursive.

Now it is shown that the set A constructed satisfies $K^A(x) \leq q(K(x)) + c_0$ for some constant c_0 and all x. This needs some facts about the sequence $y_0, y_1, \ldots$ and the complexities of these strings relative to A.

For ease of notation, U^A denotes the universal prefix-free machine relative to A and $U = U^\emptyset$ the unrelativized one. Let a_n be an input of shortest length such that $U^A(a_n) = y_n$ and let b_n be an input of length y_{n-1} such that $U(b_n) = y_n$.

Now consider all the n such that $|a_n| \leq y_{n-1} - 2y_{n-2}$. Then one has a prefix-free machine V^A and a partial-recursive coding function θ such that

- $V^A(b_{n-1}a_n)$ computes $\Omega_{y_n} \upharpoonright y_{n-1} - y_{n-2} - c_1$;
- $U(\theta(b_{n-1}\Omega \upharpoonright y_{n-1} - y_{n-2} - c_1))$ computes $\min\{s : \Omega_s \upharpoonright (y_{n-1} - y_{n-2} - c_1) = \Omega \upharpoonright (y_{n-1} - y_{n-2} - c_1)\}$.

where the constant c_1 is so large that θ can be chosen such that $|\theta(b_{n-1}d)| \leq y_{n-1}$ for all $d \in \{0,1\}^{y_{n-1}-y_{n-2}-c_1}$. As a consequence, the computation $U(\theta(b_{n-1}\Omega \upharpoonright y_{n-1} - y_{n-2} - c_1))$ needs less than y_n steps. Thus, $V^A(b_{n-1}a_n)$ computes $\Omega \upharpoonright y_{n-1} - y_{n-2} - c_1$ and $|b_{n-1}a_n| = y_{n-2} + |a_n| \leq y_{n-1} - y_{n-2}$. Since Ω is random relative to A, this can happen only for finitely many n and one has that $|a_n| > y_{n-1} - 2y_{n-2}$ for almost all n.

Now assume that $n > 1$ and $|a_n| > y_{n-1} - 2y_{n-2}$. Let $E_n = \{e : U^A(e)$ needs at least $\min(I_{y_n})$ and at most $\min(I_{y_{n+1}}) - 1$ steps$\}$. Note that for $e \in E_n$, b_n is that string of length y_{n-1} for which $U(b_n)$ terminates last within the computation-time of $U^A(e)$ and $y_n = U(b_n)$. So one has a constant c_2 and for each e a prefix-free input d of length $|e| + K(y_{n-1}) + c_2$ such that $U^A(d) = y_n$. This gives that there is a constant c_3 with

$$\sum_{e \in E_n} 2^{-|e|-c_2-K(y_{n-1})} < 2^{c_3-|a_n|}$$

what using $|a_n| > y_{n-1} - 2y_{n_2}$ can be transformed to

$$\sum_{e \in E_n} 2^{y_{n-1}-c_2-c_3-3y_{n-2}-e} < 1.$$

There is a partial-recursive function g such that $g(b_n) = |I_{y_0} \cup I_{y_1} \cup \cdots \cup I_{y_n}|$. Now one can construct a prefix-free machine which on input bd with $U(b)$ being defined and $|d| = g(b_n)$ enumerates requests of weight at most 2^{-b-d} with the additional constraint that, in the case that $b = b_n$ and d is the restriction of A to $I_{y_0} \cup I_{y_1} \cup \cdots \cup I_{y_n}$, the requests are just an enumeration of the set

$$\{\langle |b_n| + g(b_n) + |e| + c_2 + c_3 + 3y_{n-2} - y_{n-1}, U^A(e)\rangle : e \in E_n\}.$$

Recall that the weight of a request $\langle i, j\rangle$ is 2^{-i}. So the sum of the weights of all requests is at most 1. Note from b_n and d one can compute $y_0, y_1, \ldots, y_n$ and A on $I_{y_0} \cup I_{y_1} \cup \cdots \cup I_{y_n}$ so that the enumeration is effective. By the inequality

$$\sum_{e \in E_n} 2^{y_{n-1}-c_2-c_3-3y_{n-2}-e} < 1.$$

from the above one has that the bound on the weight of the requests is kept. Assume that $|e| = K^A(x)$ and $U^A(e) = x$ and x is so large that $e \in E_n$ for an n satisfying that $g(b_n) \leq 2y_{n-2}$ and that n does not fall under the finitely many exceptions considered above. Then there is a request of the form $\langle |e| + g(b_n) + c_2 + c_3 + 3y_{n-2}, x\rangle$. It follows from the Kraft–Chaitin Theorem that there is a constant c_4 with $K^A(x) \leq |e| + 5y_{n-2} + c_4$ for the n with $e \in E_n$.

As for almost all n, $|a_n| > y_{n-1} - 2y_{n-2}$ and as one can compute y_n relative to A from y_{n-2} plus an upper bound on y_n, one has that for almost all n and every e with $U^A(e)$ need more than y_n steps that $|e| > y_{n-1} - 3y_{n-2} - c_5$ for some constant c_5. Since r grows slower than every unbounded and non-decreasing recursive function and $y_{n-1} - 3y_{n-2} - c_5 > y_{n-1}/2$ for almost all n, there is a constant c_6 such that $r(e) \geq r(y_n) - c_6 = y_{n-2} - c_6$ where c_6 is independent of e, n as long as $e \in E_n$. So one has that $K(U^A(e)) \leq |e| + 5r(|e|) + c_4 + 5c_6$.

One can now cover the case, $x = U^A(e)$ the finitely many x where $U^A(e)$ needs at most $\min(I_{y_{n+1}}) - 1$ steps for some of the finitely many exceptional n in the case distinction above by taking c_0 to be sufficiently much larger than $c_4 + 5c_6$ and obtains that

$$\forall x\ K(x) \leq K^A(x) + 5r(K^A(x)) + c_0 = q(K^A(x)) + c_0$$

what completes the proof. $\square$

One should note that the real difficulty of this construction stems from the fact that the constructed set has to be p-low and not only weakly p-low. For estimation functions, the construction of weakly p-low sets is quite straightforward. Note that the resulting set is not K-trivial as it is Turing complete.

Proposition 24. *Let p be an estimation function. Then there is a Turing complete r.e. set A which is weakly p-low and also satisfies the corresponding property for C: there are constants c_K, c_C such that $K(A \upharpoonright x) \leq p(K(x)) + c_K$ and $C(A \upharpoonright x) \leq p(C(x)) + c_C$ for all x.*

Proof. For defining an enumeration of A, fix a one-one enumeration $b_0, b_1, \ldots$ of the halting problem and approximations C_s, K_s to C, K. Let $A_0 = \emptyset$. At stage $s + 1$, let a_m be the mth non-element of A_s in ascending order. Now the set A_{s+1} is computed as follows.

- Let n be the minimum of all m such that one of the following conditions holds:
  - . $a_m > s$;
  - . $b_s \leq m$;
  - . $p_s(K_s(k)) - K_s(k) \leq m$ for some k with $a_m \leq k \leq s$;
  - . $p_s(C_s(k)) - C_s(k) \leq m$ for some k with $a_m \leq k \leq s$.
- Let $A_{s+1} = A_s \cup \{x : a_n \leq x \leq s\}$.

This set A satisfies the following properties:

- A is co-infinite and r.e.;
- A is Turing complete;
- $K(A \upharpoonright x) \leq p(K(x)) + c_K$ for some constant c_K and all x;
- $C(A \upharpoonright x) \leq p(C(x)) + c_C$ for some constant c_C and all x.

The first property states the obvious fact that A is r.e. by construction. The other fact that A is co-infinite needs some more thought. Assume by way of contradiction that $|\overline{A}| = m$ for some finite number m. Let $a_0, a_1, \ldots, a_{m-1}$ denote the non-elements of A in ascending order and assume that s is so large that the following conditions hold:

- if $b_t \leq m$ then $t < s$;
- for all $x \in A - A_s$ there is no $k \geq x$ and no $e \geq \min\{C(k), K(k)\}$ such that $p(e) - e \leq m$;
- if $x \leq a_{m-1} + 1$ then $x \in A \Leftrightarrow x \in A_s$.

Then one can see that the parameters $a_0, a_1, \ldots, a_{m-1}$ chosen in the definition of step s coincide with the m least non-elements of A and are just not enumerated. Furthermore, a_m is also defined as the next non-element of A_s. Note that $a_m \leq s$ as $s \notin A_s$. Now one can see that a_m is not enumerated into A_{s+1} because the n selected is larger than m: for all $m' < m$, $n \neq m'$ because otherwise $a_0, a_1, \ldots, a_{m-1}$ would not remain outside A; furthermore, $n \neq m$ as the first and second item in the conditions on s together with the facts that p_s approximates p from above and $a_m \leq s$ imply that m does not satisfy the search-conditions. So $a_m \notin A_{s+1}$ and one can show by induction that $a_m \notin A_t$ for all $t > s$, this contradicts the assumption that $|\overline{A}| = m$. Therefore, A is co-infinite.

The second property follows from the construction. If $a_0, a_1, \ldots$ are the non-elements of A in ascending order, then $b_s \leq m$ implies that $s \leq a_m$. Thus m is in the halting problem iff $m \in \{b_0, b_1, \ldots, b_{a_m}\}$ and so the halting problem is Turing reducible to A.

The third property can be seen as follows: Given x and the shortest description σ for x with respect to a fixed prefix-free universal machine, let n be the number of non-elements of A below x. Then one can construct a prefix-free machine which from input $1^n0\sigma$ first evaluates the universal machine on σ to get the value x and then searches for a stage s such that A_s contains all but n elements below x. Having this x and s, the machine outputs $A_s \upharpoonright x$. If σ and n are chosen correctly, then the output is correct. Thus one has that $K(A \upharpoonright x)$ is at most $K(x) + n + c_K$ where the constant c_K comes from translating the given prefix-free coding of $K(A \upharpoonright x)$ of length $K(x) + n + 1$ for some machine into inputs for the universal machine. Furthermore, for all sufficiently large s, $K_s(x) + n \leq p_s(K_s(x))$ as otherwise the marker a_{n-1} would move. Therefore $K(x) + n \leq p(K(x))$ and A is weakly p-low.

The fourth property can be proven analogously; here the constructed machine is not prefix-free and σ is the shortest input producing x with respect to some fixed universal plain machine, nevertheless σ and n can of course still be recovered from $1^n0\sigma$. The rest of the proof follows the previous item but is working with C in place of K. This completes the proof of the whole result. □

For any estimation function p and the above constructed $A \in \mathcal{K}[p]$, $\Omega \leq_T A$ and thus $A \notin \mathcal{M}[p]$ by Proposition 20(i) and (iii). Thus the inclusion from Proposition 20(v) is strict.

Corollary 25. *For all estimation functions p, $\mathcal{M}[p] \subset \mathcal{K}[p]$.*

Proposition 26. *For every estimation function p there is a whole Turing degree outside Δ^0_2 contained in $\mathcal{K}[p]$.*

Proof. For any estimation function p one can consider the estimation function q given as $q(n) = n + \log(p(n) - n)/2$. Then one can construct an r.e. set A as in Proposition 24 which is in $\mathcal{K}[q]$.

The set A is not recursive. Thus, due to Yates' version of the Friedberg-Muchnik Splitting Theorem [12, Theorem IX.2.4 and Exercise IX.2.5], one can construct a partial-recursive $\{0, 1\}$-valued function ψ with domain A such that $\psi^{-1}(0)$, $\psi^{-1}(1)$ form a recursively inseparable pair, that is, ψ does not have a total extension. Actually, given a one-one enumeration $a_0, a_1, \ldots$ of A, this function ψ can be inductively defined on this domain by taking $\psi(a_s)$ in $\{0, 1\}$ such that $\psi(a_s)$ differs from $\varphi_{e,s}(a_s)$ for the least e where either $e = s$ or $\varphi_{e,s}(a_s)$ is defined and $\psi(a_t) = \varphi_{e,s}(a_t)$ for all $t < s$ with $a_t \in \mathrm{domain}(\varphi_{e,s})$.

Every total extension B of ψ is in $\mathcal{K}[p]$ as given any n and any x, the number m of places below x where ψ is undefined satisfies $m < q(K(x)) - K(x)$. Let $x_1, x_2, \ldots, x_m$ be these places. Let σ be the shortest input such that the universal machine for K computes x. Then one can code $B \upharpoonright x$ by $1^m0B(x_1)B(x_2)\ldots B(x_m)\sigma$ and thus has that $K(B \upharpoonright x)$ is below $p(K(x))$. As one can take B to have hyperimmune-free Turing degree [12, Theorem V.5.34] and as $\mathcal{K}[p]$ is closed under wtt-reducibility, one has that a whole Turing degree outside Δ^0_2 is contained in $\mathcal{K}[p]$. □

Note that the above result also holds with C in the place of K, the proof is exactly the same. So given an estimation function p, one can construct a hyperimmune-free Turing degree only consisting of sets E satisfying $C(E \upharpoonright x) \leq p(E(x))$ for all x up to an additive constant. Unfortunately, it is not guaranteed that this degree is also strongly jump-traceable, it is even a bit unlikely, as only the use of total E-recursive functions but not of the jump is recursively bounded in the case of a set E of hyperimmune-free degree.

Acknowledgements

The first author was partially supported by Fundación YPF. The second author was partially supported by the Marsden Fund of New Zealand grant no. UOA-319. The third author was partially supported by NUS grant number R252–000–212–112.

Major parts of this paper were written while S. Figueira and F. Stephan visited New Zealand with the support of Marsden Fund.

References

[1] Mark Bickford, Charlie F. Mills, Lowness properties of r.e. sets, Manuscript, UW Madison, 1982.

[2] Gregory Chaitin, A theory of program size formally identical to information theory, Journal of the Association for Computing Machinery 22 (1975) 329–340.
[3] Gregory Chaitin, Information-theoretical characterizations of recursive infinite strings, Theoretical Computer Science 2 (1976) 45–48.
[4] Peter Cholak, Rod Downey, Noam Greenberg, Strongly jump-traceability I: The computably enumerable case, Advances in Mathematics (in press).
[5] Rod Downey, Denis Hirschfeldt, Joseph Miller, André Nies, Relativizing Chaitin's halting probability, Journal of Mathematical Logic 5 (2) (2005) 167–192.
[6] Rod Downey, Denis Hirschfeldt, André Nies, Frank Stephan, Trivial reals, in: Proceedings of the 7th and 8th Asian Logic Conferences, World Scientific, River Edge, NJ, 2003, pp. 103–131.
[7] Santiago Figueira, Frank Stephan, Guohua Wu, Randomness and universal machines, in: CCA 2005, Second International Conference on Computability and Complexity in Analysis, in: Informatik Berichte, vol. 326, Fernuniversität Hagen, July 2005, pp. 103–116.
[8] Ming Li, Paul Vitányi, An Introduction to Kolmogorov Complexity and its Applications, second ed., Springer, Heidelberg, 1997.
[9] Jeanleah Mohrherr, A refinement of low *n* and high *n* for the r.e. degrees, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 32 (1) (1986) 5–12.
[10] André Nies, Lowness properties and randomness, Advances in Mathematics 197 (2005) 274–305.
[11] André Nies, Reals which compute little, in: Z. Chatzidakis, P. Koepke, W. Pohlers (Eds.), Proceedings of Logic Colloquium 2002, in: Lecture Notes in Logic, vol. 27, 2002, pp. 261–275.
[12] Piergiorgio Odifreddi, Classical Recursion Theory, vol. 1, North-Holland, Amsterdam, 1989; vol. 2, Elsevier, Amsterdam 1999.
[13] Sebastiaan Terwijn, Domenico Zambella, Algorithmic randomness and lowness, The Journal of Symbolic Logic 66 (2001) 1199–1205.