



# Strong jump-traceability I: The computably enumerable case<sup>☆</sup>

Peter Cholak<sup>a</sup>, Rod Downey<sup>b</sup>, Noam Greenberg<sup>b,\*</sup>

<sup>a</sup> *Department of Mathematics, University of Notre Dame, Notre Dame, IN, USA*

<sup>b</sup> *School of Mathematics, Statistics and Computer Science, Victoria University, PO Box 600, Wellington, New Zealand*

Received 4 March 2007; accepted 17 September 2007

Available online 24 October 2007

Communicated by Michael J. Hopkins

---

## Abstract

Recent investigations in algorithmic randomness have lead to the discovery and analysis of the fundamental class  $\mathcal{K}$  of reals called the  $K$ -trivial reals, defined as those whose initial segment complexity is identical with that of the sequence of all 1's. There remain many important open questions concerning this class, such as whether there is a combinatorial characterization of the class and whether it coincides with possibly smaller subclasses, such as the class of reals which are not sufficiently powerful as oracles to cup a Turing incomplete Martin–Löf random real to the halting problem. Hidden here is the question of whether there exist proper natural subclasses of  $\mathcal{K}$ . We show that the combinatorial class of computably enumerable, *strongly jump-traceable* reals, defined via the jump operator by Figueira, Nies and Stephan [Santiago Figueira, André Nies, Frank Stephan, Lowness properties and approximations of the jump, *Electr. Notes Theor. Comput. Sci.* 143 (2006) 45–57], is such a class, and show that like  $\mathcal{K}$ , it is an ideal in the computably enumerable degrees. This is the first example of a class of reals defined by a “cost function” construction which forms a *proper* subclass of  $\mathcal{K}$ . Further, we show that every c.e., strongly jump-traceable set is not Martin–Löf cuppable, thus giving a combinatorial property which implies non-ML cuppability.

© 2007 Elsevier Inc. All rights reserved.

*Keywords:* Strongly jump-traceable;  $K$ -trivial; Martin–Löf random; Computability; Computably enumerable; Turing degree

---

<sup>☆</sup> The first author's research was supported by the National Science Foundation under grant no. DMS 02-45167. The second and third authors were supported by the Marsden Fund of New Zealand, the third by a postdoctoral fellowship.

\* Corresponding author.

*E-mail addresses:* [peter.cholak.1@nd.edu](mailto:peter.cholak.1@nd.edu) (P. Cholak), [Rod.Downey@vuw.ac.nz](mailto:Rod.Downey@vuw.ac.nz) (R. Downey), [greenberg@mcs.vuw.ac.nz](mailto:greenberg@mcs.vuw.ac.nz) (N. Greenberg).

## 1. Introduction

The relationship between randomness and computational complexity has been the aim of a longstanding programme of research. Fundamental issues, for example, are the connections between the degree of algorithmic randomness of a real and its power as an oracle for computations; and the investigation of relative randomness using computability-theoretic tools. For example, random reals ought to have initial segments which are hard to compute/compress. We can ask: are they useful as oracles? The answer has been emerging in recent years. Independently Kučera and Gács proved that every real is computable from a random one, but work of Stephan, Miller and others has demonstrated that such computationally clever reals are really atypical and with probability 1, a random real has informations arranged in a computationally useless manner. (We refer the reader to the paper of Downey, Hirschfeldt, Nies and Terwijn [8] for a general review of this program and for more details of the above results).

This paper is concerned with reals whose initial segment complexity is very simple indeed. We would expect that such reals should have very low computational power. How low? In the present paper we will attempt to clarify the relationship between reals with low computational power as measured by the halting problem *relative* to the real, and reals with low initial segment complexity as measured by Kolmogorov complexity.

The first result here is the information theoretical characterization of computability due to Loveland. Loveland [15] proved that a real  $\alpha$  is computable if and only if the sequence  $C(\alpha \upharpoonright n|n)$  is bounded (here  $C$  denotes plain Kolmogorov complexity). Thus we can characterize the complexity notion of being computable using the information theoretical notion of having initial segments of low relative algorithmic information. Loveland's result was later extended by Chaitin [2], who proved that a real  $\alpha$  is computable iff the sequence  $C(\alpha \upharpoonright n) - C(n)$  is bounded.

After the introduction of prefix-free Kolmogorov complexity  $K$  by Levin [14] and then Schnorr [28] and Chaitin [2], to capture the intensional meaning of information content, people wondered if boundedness of the sequence  $K(\alpha \upharpoonright n) - K(n)$  implied that  $\alpha$  was computable. Chaitin proved that any such real must be  $\Delta_2^0$ , i.e. computable from the halting problem. Solovay [30] gave us a surprise: there exist non-computable reals  $\alpha$  with this property. Reals  $\alpha$  such that  $K(\alpha \upharpoonright n) - K(n)$  is bounded have very surprising properties and are now called *K-trivial* reals (Downey, Hirschfeldt, Nies and Stephan [7]).

The class of  $K$ -trivial reals has turned out to be a remarkable class. As is now well known they can easily be constructed by the prototypical “cost function” construction (which is simpler than Solovay's original construction). To wit, define the *cost*, or *weight* of  $x$  at stage  $s$  as

$$c(x, s) = \sum_{x \upharpoonright n \leq s} 2^{-K_s(n)}.$$

Now define a computably enumerable set  $A = \bigcup_s A_s$  by putting  $x \searrow A_{s+1} - A_s$  if  $W_{e,s} \cap A_s = \emptyset$ ,  $x > 2e$ ,  $x \in W_{e,s}$  and  $c(x, s) < 2^{-(e+1)}$ . (That is, we will put  $x$  into  $A$  at  $s$  if it diagonalizes, and does not cost us too much.) Then this set  $A$  is a simple set which is  $K$ -trivial [7].

It is known that for each  $c$  there are only  $O(2^c)$  many reals with constant of triviality  $c$  (Zambella [33]).<sup>1</sup> In [7], Downey, Hirschfeldt, Nies and Stephan introduced the construction

<sup>1</sup> This implies Chaitin's result that they are all  $\Delta_2^0$  (Chaitin [2]).

above and showed the  $K$ -trivial reals are solutions to Post's problem in that they are Turing incomplete.

After the [7] construction appeared, it was noted that there was a distinct similarity to the construction of *Martin–Löf low* reals first found in Kučera and Terwijn [13]. Here we say that a real  $A$  is Martin–Löf low if the collection of  $A$ -random reals were exactly the 1-random reals. That is,  $A$  is so weak as an oracle that no random reals are destroyed by  $A$ . Such reals were also constructed by a cost function construction identical to the above, except that the cost this time was related to the possible effect of enumerating  $x$  into  $A_{s+1} = A_s$  on  $\mu(U^A[s])$ , the universal Martin–Löf test relative to  $A$  at stage  $s$ . There was also a similar construction of a real  $A$  low for  $K$ , meaning that for all  $\sigma$ ,  $K^A(\sigma) = K(\sigma) + O(1)$ , given in unpublished work of An.A. Muchnik. Here  $A$  is so feeble as an oracle that even Kolmogorov complexity itself remains unchanged relative to  $A$ . The general feature of all of these constructions was the existence of a computably enumerable “cost function” where for each  $x$ ,  $c(x, s + 1) \geq c(x, s)$  and  $\lim_x \lim_s c(x, s) \rightarrow 0$ .

Finally, Nies [20] and Hirschfeldt and Nies (see [7]) showed that “all is one” by proving that  $A$  is  $K$ -trivial iff  $A$  is Martin–Löf low iff  $A$  is low for  $K$ . Subsequently, it was realized that the class  $\mathcal{K}$  of  $K$ -trivial reals also coincides with the class of *bases of Martin–Löf randomness*: reals  $A$  such that there is an  $A$ -random real  $B$  with  $A \leq_T B$  (Hirschfeldt, Nies and Stephan [11]). All of these results together imply that  $\mathcal{K}$  has a very nice structure: it is a  $\Sigma_3^0$  ideal, contained in the low degrees, bounded by a  $\text{low}_2$  degree, and generated by its c.e. members.

After these results, there arose a number of other cost function constructions from the literature which seemed to be different from the  $K$ -trivial one. They include the construction of a real  $A$  which is not cuppable  $\geq_T \emptyset'$  by any incomplete Martin–Löf random real (Nies [22]) and the construction of a real  $A$  which was low for weakly 2-random tests by Downey, Nies, Weber and Yu [9].<sup>2</sup> In each of these constructions it seemed that the cost function went to zero much more slowly than cost functions associated with  $K$ . It remains an open question whether the Martin–Löf cuppable reals are exactly the  $K$ -trivials. Perhaps surprisingly, it has been shown that the reals low for weak 2-randomness (and weak 2-randomness tests) coincide with  $\mathcal{K}$  ([9] and [16,23]). Again all is one!

Related to all of this is the fundamental notion of traceability. We say that a function  $h : \omega \rightarrow \omega \setminus \{0\}$  is an *order* (Schnorr [27]) if  $h$  is computable, non-decreasing and  $\lim_s h(s) = \infty$ . We say that a function  $f : \omega \rightarrow \omega$  is *computably traceable* with respect to the order  $h$  if there is a computable sequence  $\langle F_x \rangle_{x < \omega}$  of finite sets such that for all  $x$ ,  $|F_x| \leq h(x)$  and  $f(x) \in F_x$ . We will say that a degree  $\mathbf{a}$  is computably traceable iff there is some order  $h$  such that every  $f$  of degree  $\mathbf{a}$  or less can be computably traced with respect to  $h$ . Finally, we will say that  $\mathbf{a}$  is *strongly* computably traceable iff it is computably traceable with respect to any order. Here the idea is that the real is *computationally feeble*, in the sense that we have very good approximations to computations using  $A$  as an oracle. Such reals are highly non-random.

Terwijn and Zambella [32] showed that a real  $A$  is low for Schnorr randomness tests (see [5]) iff  $\text{deg}(A)$  is computably traceable iff  $\text{deg}(A)$  is strongly computably traceable. This was extended to the randomness notions by Kjos-Hanssen, Stephan, Nies and others [12,20], and finally to the low-for-computable-machines by Downey, Greenberg, Mihailović and Nies [4]. Thus lowness related to Schnorr randomness has a “combinatorial” characterization (meaning one that does not mention Kolmogorov complexity).

<sup>2</sup> Recall that  $B$  is weakly 2-random means that  $B$  is a member of every  $\Sigma_2^0$  class of measure 1.

It is a fundamental question (see e.g. Miller and Nies [17]) whether there is a similar combinatorial characterization of  $K$ -triviality.

Zambella (see Terwijn [31]) showed that if  $A$  is  $K$ -trivial then  $\deg(A)$  is c.e. traceable. Here we define a (c.e.) *trace* to be a uniformly c.e. sequence  $\langle T_x \rangle$  of finite sets<sup>3</sup>; a trace *traces* a function  $f$  if for all  $x$ ,  $f(x) \in T_x$ ; and the tracing *obeys* an order  $h$  if for all  $x$ ,  $|T_x| \leq h(x)$ . Finally, a degree  $\mathbf{a}$  is *c.e. traceable* if there is an order  $h$  such that every  $f \leq_T \mathbf{a}$  can be traced by some trace obeying  $h$ . Thus, Zambella showed that  $K$ -triviality also *implies* at least some combinatorial property. Nies [20,21] showed that  $K$ -triviality also implies a stronger combinatorial property. He showed that all  $K$ -trivial reals were *jump-traceable* by computably enumerable sets. Here we will denote that jump  $\{e\}^X(e)$  by  $J^X(e)$ ,<sup>4</sup> and say that  $A$  is *jump-traceable* if there is some order  $h$  and a c.e. trace  $\langle T_x \rangle$  which respects (obeys)  $h$  and which traces  $J^A$ , where the requirement for tracing the partial function is that  $J^A(e) \in T_e$  if  $e \in \text{dom } J^A$ .

This result motivated Figueira, Nies and Stephan [10,21] to study the notions of jump-traceability, and the related one of strong jump-traceability. We say that  $A$  is strongly jump-traceable iff  $J^A$  can be traced obeying any order. Nies showed [21] that jump-traceability coincides on the computably enumerable sets with the notion of superlowness (that is,  $A' \equiv_{\text{IT}} \emptyset'$ ) introduced by Bickford and Mills [1] and Mohrherr [19], but the notions differed outside of the computably enumerable sets. In [10], Nies, Figueira and Stephan constructed a non-computable, strongly jump-traceable, computably enumerable real, using a construction resembling one using a cost function. They then showed that jump-traceability and strong jump-traceability differ on the computably enumerable reals.

All of this, and the quest for a combinatorial characterization of the  $\mathcal{K}$  lead Miller and Nies [17] to ask if  $\mathcal{K}$  was exactly the class of strongly jump-traceable reals. At the time, they did suggest that this is unlikely.

In this paper, we will clarify the situation for computably enumerable reals.

**Theorem 1.1.** *Every c.e. strongly jump-traceable set is  $K$ -trivial.*

Thus for the first time, we have an example of a combinatorial property (by which we mean here a property whose definition does not involve randomness or Kolmogorov complexity) that at least *implies*  $K$ -triviality. The proof of this result relies on a new combinatorial technique using a kind of amplification of the traceability along the lines of the decanter or golden run method. It is beyond known technology; we believe that it could have other applications within computability theory and randomness.

On the other hand we also prove the following.

**Theorem 1.2.** *There is a  $K$ -trivial c.e. set that is not strongly jump-traceable. Indeed it is not jump-traceable with a bound of size roughly  $\log \log n$ .*

This is the first example of a class defined by cost functions which we know does not coincide with  $\mathcal{K}$ . Again the proof technique is novel, since it is the first time a cost function has been used which still allows for the defeat of one involving Kolmogorov complexity.

This work leads to certain speculations. We know that if  $A$  is  $K$ -trivial, then by [10],  $A$  is jump-traceable with respect to an order roughly  $h(n) = n \log n$ . On the other hand, the proof of

<sup>3</sup> That is, there is a computable function  $g$  such that for all  $x$ ,  $T_x = W_{g(x)}$ .

<sup>4</sup> In other words, for all  $X$ ,  $J^X$  is the universal function which is partial computable in  $X$ .

Theorem 1.1 shows that if a c.e. set  $A$  is jump-traceable with respect to about  $\sqrt{\log n}$  then it is  $K$ -trivial. It seems reasonable to suggest that there might well be a combinatorial characterization of the c.e.  $K$ -trivial reals as those which are jump-traceable with respect to an order (or orders) at some critical growth value between these two extremes. It may well be that a finer analysis of the two theorems here, and of the work in [10], might allow for such a characterization.

The reader might wonder if the class of strongly jump-traceable reals coincides with the class of Martin–Löf non-cupable reals.

**Theorem 1.3.** *No c.e. strongly jump-traceable set cups over  $\emptyset'$  with a Martin–Löf random set.*

Thus also for the first time, we have a combinatorial property which implies non-ML cupability.

As mentioned above, early on, Nies [20] proved that the  $K$ -trivial reals form an ideal in the Turing degrees. For our last result we will demonstrate that the computably enumerable strongly jump-traceable degrees also form a proper sub-ideal of the  $K$ -trivial reals.

**Theorem 1.4.** *If  $A$  and  $B$  are strongly jump-traceable and c.e., then so is  $A \oplus B$ .*

Again, we prove something stronger. We prove that if  $A$  and  $B$  are c.e. and traceable via sufficiently slowly growing functions  $h_A$  and  $h_B$ , then  $A \oplus B$  is traceable via a trace computably related to  $h_A$  and  $h_B$ . Aside from its intrinsic interest, this should be compared with the result of Bickford and Mills [1] that there are two superlow c.e. sets  $X$  and  $Y$  with  $X \oplus Y$  Turing complete.

The method of proof for this last result is again novel, and uses this kind of decanter method of “infinite depth.”

In a subsequent paper [3], the last two authors investigate the case where the sets are not computably enumerable. Here the situation is less clear. Using a very difficult argument, Downey and Greenberg showed that if  $A$  is any strongly jump-traceable set then  $A$  is  $\Delta_2^0$ , with arbitrarily slow enumerations (it seems that such sets should be  $K$ -trivial, but this is still under investigation). Again this raises a question regarding the rate of growth. Nies showed [21] that for some level ( $h(n) \sim 2^{2^n}$ ), there are uncountably many sets which are jump-traceable with respect to  $h$ . For  $h(n)$  about  $\log \log n$ , however, every set which is jump-traceable with respect to  $h$  is  $\Delta_2^0$  and hence there are only countably many such sets.

We remark that our results suggest the following problem: *Is  $A$   $K$ -trivial iff for all orders  $h$  with  $\sum_{n \in \mathbb{N}} 1/h(n) < \infty$ ,  $A$  is jump-traceable with order  $h$ ?*

### 1.1. Notation and basic facts and definitions

This paper is concerned with prefix-free Kolmogorov complexity, which we will denote by  $K$ ;  $U$  is the universal prefix-free machine, and  $U^X$  is the universal oracle prefix-free machine. We will refer to “reals” which will be identified with Cantor space  $2^\omega$ . The initial segment of length  $n$  of a real  $A$  will be denoted by  $A \upharpoonright n$ . For every  $X \in 2^{\leq \omega}$  we let  $\Omega^X = \mu(\text{dom } U^X)$ . Notation will be standard in the sense that we will follow [6,8,17,20,24]. Notation for the computability used follows Soare [29] (and also [25,26]) unless specifically noted otherwise.

Recall again that an *order function* is a computable, non-decreasing and unbounded function  $h : \omega \rightarrow \omega \setminus \{0\}$ .

Let  $\langle \Psi_c \rangle_{c < \omega}$  be an enumeration of all Turing functionals which compute partial functions from an oracle. There is a uniformly computable sequence  $\langle \alpha_c \rangle$  of order functions such that for all  $X \in 2^\omega$ , each  $\alpha_c$  reduces  $\Psi_c^X$  to  $J^X$ , that is,

$$\Psi_c^X = J^X \circ \alpha_c.$$

We may assume that  $\langle \alpha_c(0) \rangle_{c < \omega}$  is strictly increasing.

**Lemma 1.5.** *Let  $\langle h_c \rangle$  be a uniformly computable sequence of order functions, such that  $\langle h_c(0) \rangle_{c < \omega}$  is non-decreasing and unbounded. Then there is an order function  $\tilde{h}$  such that for all  $x$  and  $c$ ,*

$$\tilde{h}(\alpha_c(x)) \leq h_c(x).$$

**Proof.** For all  $y$  and  $c$ , let  $l(y, c)$  be the least  $x$  such that  $\alpha_c(x) \geq y$ ; and let  $\tilde{h}(y) = \min_{c < \omega} h_c(l(c, y))$ . This is computable because for large enough  $c$  we have  $l(c, y) = 0$  and  $\langle h_c(0) \rangle$  is non-decreasing.  $\square$

Again recall that a *trace* is a uniformly c.e. array of finite sets. A trace  $\langle T_x \rangle_{x < \omega}$  *traces* a partial function  $p$  if for all  $x \in \text{dom } p$ ,  $p(x) \in T_x$ . A trace *obeys* an order  $h$  if for all  $x < \omega$ ,  $|T_x| \leq h(x)$ . A set  $A \subset \omega$  is *strongly jump-traceable* if for every order  $h$ , there is a trace for  $J^A$  which obeys  $h$ .

**Lemma 1.6.** *A set  $A$  is strongly jump-traceable iff for every function  $p$  which is partial computable in  $A$  and for every order  $h$ , there is a trace for  $p$  which obeys  $h$ .*

This lemma (which is implicit in [10]) shows that the collection of strongly jump-traceable sets is downwards closed under Turing reduction (and in particular is degree invariant).

**Proof.** Suppose that  $p = \Psi_c^A$ , and let  $h$  be an order. By a simplified Lemma 1.5, there is an order  $\tilde{h}$  such that for all  $x$ ,  $(\tilde{h} \circ \alpha_c)(x) \leq h(x)$ . From a trace for  $J^A$  which obeys  $\tilde{h}$  we can get a trace for  $p$  which obeys  $h$ .  $\square$

## 2. A $K$ -trivial set which is not strongly jump-traceable

The construction of a  $K$ -trivial set which is not strongly jump-traceable came out of a direct construction of a  $K$ -trivial set which is not  $n$ -c.e. for any  $n$ . As mentioned in the introduction, the collection of  $K$ -trivial sets is closed downwards under Turing reduction, and so it must contain sets that are not  $n$ -c.e. for any  $n$ . But how would a direct construction of such a set go?

By [20] we know that any construction will essentially be a cost-function construction, such as the by now classic construction of a promptly simple, c.e.  $K$ -trivial set mentioned in the introduction. That construction can be redescribed as follows. The  $e$ th requirement  $R_e$  wishes to show that the set  $A$  we construct is not co-c.e. via the  $e$ th co-c.e. approximation, namely  $\bar{W}_e$ . The requirement is given the sum of  $2^{-e}$  which is the capital it is allowed to spend. It appoints a follower  $x_0$ , and waits for its realisation, that is, for  $x_0 \notin \bar{W}_e$ . If, upon realisation, the cost of changing  $A(x_0)$  is greater than  $2^{-e}$ , the follower is abandoned, a new one  $x_1$  is picked, and the process repeats itself.

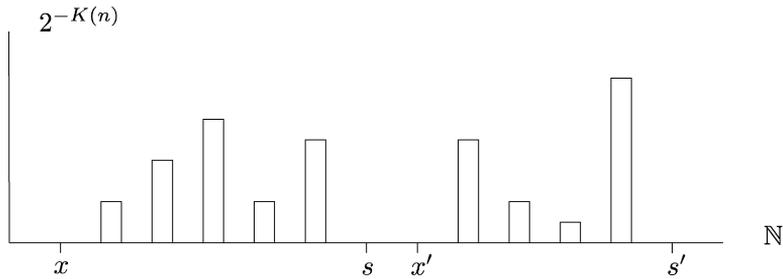


Fig. 1. The cost of enumerating  $x$  into  $A$  at stage  $s$  is  $\sum_{x < n < s} 2^{-K_s(n)}$  and the cost of enumerating  $x'$  into  $A$  at stage  $s'$  is  $\sum_{x' < n < s'} 2^{-K_{s'}(n)}$ ; the weights are “disjoint.”

Suppose now that we want to ensure that the constructed set  $A$  is not 2-c.e. The  $e$ th requirement wants to ensure that  $A$  is not 2-c.e. via the  $e$ th 2-c.e. approximation  $X_e = W_{e_0} \setminus W_{e_1}$ . Again the requirement is provided with  $2^{-e}$  much capital to spend. It may appoint a follower  $x_0$  and wait for first realisation, namely  $x$  enters  $X_e$ . Provided the price is not too high, the requirement would then extract  $x_0$  from  $A$  (we start with  $A = \omega$ ) and wait for second realisation, i.e.  $x$  leaving  $X_e$ . It would then wish to re-enumerate  $x_0$  into  $A$  and thus confirm a win on the requirement. The point here is that the follower needs two “permissions” from the cost-function, and the danger is that we spend some capital on the first action (the extraction), but the second action would be too expensive and the follower would have to be abandoned. The amount we spent on extraction is non-refundable, though, and so this strategy would soon run into trouble.

A better strategy is the following. From the initial sum  $2^{-e}$ , set aside a part (say  $2^{-(e+1)}$ ) which is kept for the second re-enumeration of a follower and will not be used otherwise (for extraction). Of the remaining  $2^{-(e+1)}$ , we apportion some (say  $2^{-(e+2)}$ ) for the sake of extraction of the first follower  $x_0$ . If the cost of extraction of  $x_0$  is higher, then we abandon  $x_0$  (at no cost to us) and allot the same amount  $2^{-(e+2)}$  for the extraction of the next follower  $x_1$ . Suppose, for example, that we did indeed extract  $x_1$ , but when it is realised again and we are ready to re-enumerate it into  $A$ , its cost has risen beyond the sum  $2^{-(e+1)}$  which we set aside for this task. We have to abandon  $x_1$ , appoint a new follower  $x_2$ , and start from the beginning. We did lose an uncompensated  $2^{-(e+2)}$ ; so we reduce the sum that we may spend on extracting  $x_2$  to  $2^{-(e+3)}$ , and keep going.

Between extractions, the sum we may spend on the next extraction is kept constant, and so the usual argument shows that some future follower will get extracted (all this assuming that all followers are realised, of course). On top of this, abandoning followers upon re-enumeration may happen only finitely many times, because each such abandoned follower  $x$  carries a cost of  $2^{-(e+1)}$  which comes from descriptions of numbers below the stage at which that follower is abandoned. The next follower  $x'$  is appointed only after the previous one is cancelled, and is chosen to be large; the cost associated with  $x$  will not be counted toward changing  $A(x')$ , and so if  $x'$  is abandoned upon re-enumeration, this is due to a completely different part of the universal machine which has weight of at least  $2^{-(e+1)}$  (see Fig. 1). We can thus see that the process cannot happen more than  $2^{e+1}$  many times.

In fact, we note that the same reasoning may be applied to the extraction steps; new followers are chosen large after we abandon a previous follower upon extraction, and since between extractions the acceptable price is fixed at some  $2^{-m}$ , this kind of abandonment will not happen (between extractions) more than  $2^m$  times. Inductively, we can determine in advance a bound on

the number of possible failures, and if we wish, we can distribute the permissible costs evenly, as we do below.

Finally, for  $n > 2$ , we apply this strategy with  $n$  layers of apportioning pieces of capital to various attempts at changing  $A(x)$  on some follower  $x$ ,  $n$  many times. To make  $A$  not be strongly jump-traceable rather than not  $n$ -c.e., what we need to do is to change  $J^A(x)$  on some input  $x$  more than  $h(x)$  many times, where  $h$  is some order we will specify in advance (and  $x$  is a “slot” in the jump that we control). To change  $J^A(x)$  we need to put the use of this computation into  $A$ ; keeping  $A$  c.e., this means that we change the use, but the principle that the same  $x$  receives attention  $h(x)$  many times remains and so the same strategy works.

2.1. The formal construction and proof of Theorem 1.2

We enumerate a set  $A$  and a function  $p$ , partial computable in  $A$ . The requirement  $R_e$  is that  $\langle W_e^{[x]} \rangle_{x < \omega}$  is not a trace for  $p$  which obeys an order function  $h$ , which we soon define. By Lemma 1.6, this will suffice to show that  $A$  is not strongly jump-traceable.

For  $e < \omega$ , let  $T_e$  consist of all sequences  $\langle k_0, k_1, \dots, k_i \rangle$  where  $i < e$  and for each  $j \leq i$  we have  $k_j < 2^{e2^j}$ . Note that indeed  $T_e$  is a tree, i.e., is closed under taking initial segments. A node  $\sigma \in T_e$  is a leaf of  $T_e$  iff it has length  $e$ . If  $\sigma \in T_e$  is not a leaf, then we let  $\epsilon_\sigma = \epsilon_\sigma^e = 2^{-e2^{|\sigma|}}$ .

The idea here is that each leaf on the tree corresponds to a particular attempt at meeting  $R_e$ , and if  $\sigma \in T_e$  is a leaf, and  $i < e$ , then  $\epsilon_\sigma \upharpoonright i$  is the amount that we would be willing to spend on the  $(e - i)$ th attack with the follower corresponding to  $\sigma$ . The tree  $T_e$  and the rationals  $\epsilon_\sigma$  were chosen so that:

- (1)  $\epsilon_\emptyset = 2^{-e}$ ;
- (2) if  $\sigma \in T_e$  is not a leaf, then it has exactly  $1/\epsilon_\sigma$  many immediate successors on  $T_e$ ; and further,
- (3) if  $|\sigma| < e - 1$  then the sum of  $\epsilon_\tau$ , as  $\tau$  ranges over immediate successors of  $\sigma$  on  $T_e$ , is  $\epsilon_\sigma$ .

These facts let us, by reverse induction on  $|\sigma|$ , show that for  $\sigma \in T_e$  which is not a leaf, the sum of  $\epsilon_\tau$ , as  $\tau$  ranges over all extensions of  $\sigma$  on  $T_e$  which are not leaves, is  $(e - |\sigma|)\epsilon_\sigma$ . Thus the sum of  $\epsilon_\tau$ , as  $\tau$  ranges over all nodes on  $T_e$  which are non-leaves, is  $e2^{-e}$ . This will be the total amount we let  $R_e$  spend; and so the construction will obey the cost-function, as  $\sum_{e < \omega} e2^{-e}$  is finite.

We can now define  $h$ . Partition  $\omega$  into intervals  $\langle I_e \rangle$  (so  $\max I_e + 1 = \min I_{e+1}$ ), letting the size of  $I_e$  be the number of leaves of  $T_e$ ; we index the elements of  $I_e$  as  $x_\sigma$  for leaves  $\sigma$  of  $T_e$ . We define  $h(x) = e - 1$  for all  $x \in I_e$ .

Note that the size of  $T_e$  is of the order of  $2^{2^e}$ , which means that  $h$  grows roughly like  $\log \log x$ .

The requirements  $R_e$  act independently. If not yet satisfied at stage  $s$ , the requirement  $R_e$  will have a pointer  $\sigma = \sigma_e[s]$  pointing at some leaf of  $T_e$ ; the requirement will be conducting an attack with  $x_\sigma$  at some level  $i < e$  (the level will be decreasing with time, until the attack is abandoned, or fully succeeds when we get to the root).

In the beginning, we let  $\sigma[0] = 0^e$ , the leftmost leaf of  $T_e$  (we order the nodes of  $T_e$  lexicographically); and we begin an attack with  $x_{\sigma[0]}$  on level  $e - 1$ .

The following are the instructions for an attack on level  $i < e$  (at a stage  $s$ ). Let  $\sigma = \sigma[s]$ . Recall that the cost of enumerating a number  $x$  into  $A$  at stage  $s$  is

$$c(x)[s] = \sum_{n \in (x, s)} 2^{-K(n)[s]}.$$

- (1) Define  $p^A(x_\sigma) = s$  with use  $s + 1$ . Wait for  $s \in W_e^{[x_\sigma]}$ . [While waiting, if some other requirement puts a number  $y \leq s$  into  $A$  and so makes  $p(x_\sigma)$  undefined, redefine  $p(x_\sigma)$ , again with value  $s$  and use  $s + 1$ .]
- (2) At stage  $t > s$ ,  $s$  enters  $W_e^{[x_\sigma]}$ . Compare the cost  $c(s)[t]$  of putting  $s$  into  $A$  at this stage with the permissible waste  $\epsilon_{\sigma \upharpoonright i}$ .
  - If  $c(s)[t] \leq \epsilon_{\sigma \upharpoonright i}$ , then enumerate  $s$  into  $A$  (making  $p(x_\sigma)$  undefined). Leave  $\sigma$  unchanged and attack with it on level  $i - 1$ . If already  $i = 0$  then declare victory and cease all action.
  - If  $c(s)[t] > \epsilon_{\sigma \upharpoonright i}$  then we abandon  $x_\sigma$ . Move one step to the right of  $\sigma \upharpoonright i + 1$ . That is, if  $\sigma = (k_0, \dots, k_{e-1})$  then let

$$\sigma[t + 1] = (k_0, \dots, k_{i-1}, k_i + 1, 0, \dots, 0).$$

Attack with the new  $\sigma$  on level  $e - 1$ .

*Justification*

We must argue that the above algorithm is consistent: in this case, that if at some stage  $t$  we want to abandon an attack with  $x_\sigma$  on level  $i < e$  and redefine  $\sigma[t + 1]$ , then the string we defined above is actually on  $T_e$ , which will hold iff  $k_i + 1 < 1/\epsilon_{\sigma \upharpoonright i}$ .

Fix such an  $i$  and  $\sigma$ . Let  $\sigma^* = \sigma \upharpoonright i$  and let  $m = \sigma(i)$ . We know that for all  $k \leq m$ , some attack was made with some string extending  $\sigma^* \frown k$  (for example with  $\sigma^* \frown k \frown (0, \dots, 0)$ ); let  $\tau_k$  be the rightmost string extending  $\sigma^* \frown k$  which was ever used for an attack (so  $\tau_m = \sigma$ ); so we know that we attacked with  $\tau_k$  on level  $i$  and that this attack is abandoned. Let  $s_k$  be the stage at which the attack with  $\tau_k$  on level  $i$  began, and let  $t_k > s_k$  be the stage at which this attack was abandoned (so  $t_m = t$ ).

The key point, as discussed above, is that  $t_{k-1} \leq s_k$ , so the intervals  $(s_k, t_k)$  are disjoint. At stage  $t_k$ , the attack with  $\tau_k$  is abandoned because  $c(s_k)[t_k] > \epsilon_{\sigma^*}$ . Now

$$\begin{aligned} 1 > \mu(\text{dom } U) &> \sum_{k \leq m} \sum_{n \in (s_k, t_k)} 2^{-K(n)} \geq \sum_{k \leq m} \sum_{n \in (s_k, t_k)} 2^{-K(n)[t_k]} \\ &= \sum_{k \leq m} c(s_k)[t_k] > (m + 1)\epsilon_{\sigma^*}. \end{aligned}$$

It follows that  $m + 1 < 1/\epsilon_{\sigma^*}$  as required.

*Verification*

First, note that by the instructions given, for each  $e < \omega$ , for each  $\tau \in T_e$  which is not a leaf, there is at most one  $s < \omega$  which is enumerated into  $A$  because of a successful attack with some  $\sigma \supset \tau$  on level  $|\sigma|$ . Thus  $R_e$  did not spend more than  $e2^{-e}$  and so the construction obeys the cost function, making  $A$   $K$ -trivial.

Fix  $e < \omega$ . There are two possible outcomes for  $R_e$ .

- (1) There is some stage  $s$  at which we begin an attack with  $x_{\sigma[s]}$  at some level, but  $s$  never turns up in  $W_e^{[x_\sigma]}$ . The attack is never concluded. But in this case, no further modifications are made for  $p(x_\sigma)$  and it has a final value  $s$ , which is not traced.

(2) Some attack with some  $x_\sigma$  on level 0 succeeds. This means that

$$|W_e^{[x_\sigma]}| \geq e > h(x_\sigma)$$

and so the trace does not obey the order  $h$ .

In either case, we see that  $R_e$  is met, and so  $A$  is not strongly jump-traceable.

**3. The computably enumerable, strongly jump-traceable degrees form a non-principal ideal**

As we mentioned above, Figueira, Nies and Stephan [10] showed that the strongly jump-traceable sets are downward closed under Turing reduction. In this section we show that the join of two c.e., strongly jump-traceable sets is also strongly jump-traceable, and so in the c.e. degrees, the strongly jump-traceable degrees form an ideal. In fact, we show that for every order function  $g$  there is another order function  $f$  such that if sets  $A_0$  and  $A_1$  are c.e. and jump-traceable via  $f$ , then  $A_0 \oplus A_1$  is jump-traceable via  $g$ .

The construction is the simplest known example of the *box amplification* (or *promotion*) method, and so we wish to describe the motivation for its discovery. For this, we need to examine the construction of a non-computable, strongly jump-traceable real.

For simplicity, suppose that  $h$  is a slow-growing order, and that we wish to construct a non-computable c.e. set  $A$  which is jump-traceable with respect to  $h$ . Let  $P_e$  be the  $e$ th non-computability requirement (which says that  $A \neq \bar{W}_e$ ) and let  $N_e$  be the requirement which is responsible for enumerating that part  $T_e$  of the trace we build which is supposed to trace  $J^A(e)$ .

We order the requirements thus:

$$\underbrace{N_0 N_1 N_2 \cdots N_e \cdots}_{h(e)=1} P_0 \underbrace{\cdots N_e \cdots}_{h(e)=2} P_1 \underbrace{\cdots N_e \cdots}_{h(e)=3} P_2 \cdots \tag{3.1}$$

The construction is now straightforward: each  $P_e$  is appointed a follower  $x$ . If at stage  $s$ ,  $P_e$  is not yet satisfied, and  $x$  appears in  $W_e$ , then it is enumerated into  $A$ , and  $P_e$  becomes satisfied. If a new computation  $J^A(s)$  appears at stage  $s$ , then  $N_e$  traces its value in  $T_e$  and initialises all weaker positive requirements, which will need to be appointed new, large followers.

The key to the success of this construction is that each requirement  $P_e$  acts at most once, and does not need to act again even if it is initialised. It may be instructive to think of the priority ordering as dynamic; when  $P_e$  acts, then it is removed from the list of requirements and is never troubled (nor does it influence other requirements) again.

To make  $A$  jump-traceable via all orders  $h$ , a further dynamic element is introduced to the priority ordering. The property of a partial computable function being an order function is  $\Pi_2^0$ , and we approximate it in this fashion. Say that a stage  $s$  is *e-expansionary* if at this stage we have further evidence that the  $e$ th partial computable function  $\varphi_e$  is an order function. If the stage is indeed *e-expansionary* then the positive requirements are pushed down the ordering so that for every  $x$  such that  $\varphi_e(x) \downarrow [s]$ , there are at most  $\varphi_e(x)$  many positive requirements stronger than  $N_{e,x}$ , the requirement that traces  $J^A(e)$  with at most  $\varphi_e(x)$  many values. To protect the positive requirements from being moved down infinitely often, we insist that a positive requirement  $P_{e'}$  cannot be moved by  $\varphi_e$  if  $e' < e$ ; these positive requirements are ignored when we count

the number of positive requirements which appear before some  $N_{e,x}$ . If  $P_{e'}$  acts then we initialise every  $N_{e,x}$  and start a new trace.

All this still allows us to use Robinson’s trick. We can prove:

**Theorem 3.1.** *If  $B$  is a low, c.e. set, then there is some strongly jump-traceable c.e. set  $A$  which is not computable in  $B$ .*

This would show that the ideal of c.e., strongly jump-traceable degrees, is not principal (as they are all low).

**Sketch of proof.** For simplicity, we fix a slow-growing order  $h$  and sketch the enumeration of some c.e. set  $A$ , not computable from  $B$ , which is jump-traceable obeying  $h$ ; for a strongly jump-traceable set, we complicate the current construction as before. Let  $P_e$  now stipulate that  $A \neq \Phi_e(B)$ . The requirement appoints a follower  $x$ , and tries to enumerate it into  $A$  at a stage at which it seems like  $\Phi_e(B, x) \downarrow = 0$ . By the recursion theorem, we have, at stage  $s$ , an approximation for the answer to the question “does the requirement  $P_e$  ever ask about a follower  $x$  which is realised by a  $B$ -correct computation?”; the requirement only enumerates  $x$  into  $A$  if it believes that the answer to the question is “yes.”

Again, after acting, the requirement  $P_e$  is removed from the list. If, though, at a later stage  $t$ , we see that the computation realising the follower  $x$  and which was believed at stage  $s$ , is actually incorrect, then  $P_e$  needs to be resuscitated. It is brought back from the dead and is placed in the place of some weaker requirement (pushing the rest further down, to maintain having just one positive requirement between blocks of negative requirements). As the guesses eventually stabilise, this cannot happen infinitely often.  $\square$

We return to the join theorem. Suppose that we wanted to prove the theorem wrong, that is, to construct c.e. sets  $A_0$  and  $A_1$  which are strongly jump-traceable but such that  $A_0 \oplus A_1$  is not. We would presumably attempt to use the strategy of Section 2 and try to diagonalise against possible traces for  $\Phi^{A_0 \oplus A_1}$  by changing its values sufficiently many times, this time by enumerating the current use into either  $A_0$  or  $A_1$ . In the priority ordering of the requirements we place both these diagonalisation requirements, and the requirements which try to trace  $J^{A_0}$  and  $J^{A_1}$  as in the construction of a strongly jump-traceable c.e. set.

Again recall that in this construction, after some requirement  $P_e$  acts, it gets removed from the list, and the blocks of  $N_x$  requirements to its left and to its right are merged; in a sense, this increases the priority of those to the right, because they suffered an injury—which means that the number of times they can be injured has just decreased by one. They have been *promoted*.

In our false construction, suppose we start with the same ordering (except that there are two kinds of negative requirements, one for  $A_0$  and one for  $A_1$ ). Each time a positive requirement  $P_e$  acts, and say enumerates a number into  $A_0$ , it needs to be demoted down the list and placed after all the negative requirements it has just injured; since these requirements may later impose new restraint, a new follower for  $P_e$  may be needed each time one such requirement decides to impose restraint. Since some of the negative requirements are also promoted by positive requirements weaker than  $P_e$ , we cannot put any computable bound, in advance, on the last place of  $P_e$  on the list, and hence, on the number of followers it will need. Thus we cannot state the computable bound which we mean to beat, and the construction fails.

This failure is turned around into our proof. Now we are given two c.e., strongly jump-traceable sets  $A_0$  and  $A_1$ , and an order function  $g$ , and we wish to trace  $J^{A_0 \oplus A_1}$ , obeying  $g$ .

Fix an input  $e$  (the requirements that trace  $J^{A_0 \oplus A_1}$  act completely independently). When at some stage of the construction we discover that  $J^{A_0 \oplus A_1}(e)$  converges, before we trace the value, we want to receive some confirmation that this value is genuine. Say that the computation has use  $\sigma_0 \oplus \sigma_1$ , where  $\sigma_i \subset A_i[s]$ . What we do is define functionals  $\Phi_0$  and  $\Phi_1$ , and define  $\Phi_i^{\sigma_i}(x) = \sigma_i$ . If indeed  $\sigma_i \subset A_i$  then  $\sigma_i$  would appear as a value in a trace  $T_x^i$  for  $\Phi_i^{A_i}$  which we receive (using the universality of  $J^{A_i}$  and the recursion theorem). Thus we can wait until both strings  $\sigma_i$  appear in the relevant “box”  $T_x^i$ , and only then believe the computation  $J^{A_0 \oplus A_1}(e)[s]$ . Of course, it is possible that both  $\sigma_i$  appear in  $T_x^i$  but that neither  $\sigma_i$  is really an initial segment of  $A_i$ ; in which case we will have traced the wrong value. In this case, however, both boxes  $T_x^i$  have been promoted, in the sense that they contain an element ( $\sigma_i$ ) which we know is not the real value of  $\Phi_i^{A_i}(x)$ , and  $\Phi_i^{A_i}(x)$  becomes undefined (when we notice that  $A_i$  moved to the right of  $\sigma_i$ ) and is therefore useful for us for testing another potential value of  $J^{A_0 \oplus A_1}(e)$  which may appear later. If the bound on the size of  $T_x^i$  (which we prescribe in advance, but has to eventually increase with  $x$ ) is  $k$ , then we originally think of  $T_x^i$  as a “ $k$ -box,” a box which may contain up to  $k$  values; after  $\sigma_i$  appears in  $T_x^i$  and is shown to be wrong, we can think of the promoted box as a  $(k - 1)$ -box. Eventually, if  $T_x^i$  is promoted  $k - 1$  many times, then we have a 1-box; if a string  $\sigma_i$  appears in a 1-box then we know it must be a true initial segment of  $A_i$ . In this way we can limit the number of false  $J^{A_0 \oplus A_1}(e)$  computations that we trace. Since all requirements act independently, this allows us to trace  $J^{A_0 \oplus A_1}$  to any computable degree of precision we may like.

That is the main idea of all “box-promotion” constructions. Each construction is infused with combinatorial aspects which counter difficulties that arise during the construction (difficulties which we think of as possible plays of an opponent, out to foil us). The combinatorics determine how slowly we want the size of the given trace to grow, and which boxes should be used in every test we make. In this construction, the difficulty is the following: in the previous scenario, it is possible, say, that  $\sigma_0$  is indeed a true initial segment of  $A_0$ , but  $\sigma_1$  is not an initial segment of  $A_1$ . And to make matters worse, the latter fact is discovered even before  $\sigma_1$  turns up in  $T_x^1$ . However, we already defined  $\Phi_0^{A_0}(x) = \sigma_0$  with  $A_0$ -correct use, which means that the input  $x$  will not be available later for a new definition. The box  $T_x^0$  has to be discarded, and further, we got no compensation—no other box has been promoted. As detailed below, the mechanics of the construction instruct us which boxes to pick so that this problem can in fact be countered. The main idea (which again appears in all box-promotion constructions) is to use clusters of boxes (or “meta-boxes”) rather than individual boxes. Instead of testing  $\sigma_i$  on a single  $T_x^i$ , we bunch together a finite collection  $M_i$  of inputs  $x$ , and define  $\Phi_i^{\sigma_i}(x) = \sigma_i$  for all  $x \in M_i$ . We only believe the computation  $J^{A_0 \oplus A_1}(e)$  if  $\sigma_i$  has appeared in  $T_x^i$  for all  $x \in M_i$ . If this is believed and then later discovered to be false, then all of the boxes included in  $M_i$  have been promoted; we can then break  $M_i$  up into smaller meta-boxes and use each separately; thus we magnify the promotion, to compensate for any losses we may occur on the other side.

### 3.1. The formal construction and proof of Theorem 1.4

In what follows, we fix a number  $e$  and show how to trace  $J^{A_0 \oplus A_1}(e)$  limiting the errors to a prescribed number  $m$ . To do this, given the number  $m$ , the requirement will ask for an infinite collection of boxes, and describe precisely how many  $k$ -boxes, for each  $k$ , it requires for its use (for  $A_0$  and  $A_1$ ). As  $m$  grows, the least  $k$  for which  $k$ -boxes are required will grow as well (we denote that number by  $k^*(m)$ ). For  $m$  and  $k \geq k^*(m)$ , let  $r(k, m)$  be the number of  $k$ -boxes which

is required to limit the size of the trace for  $J^{A_0 \oplus A_1}(e)$  by  $m$ . (In fact, if  $k \geq k^*(m), k^*(m')$  then we will actually have  $r(k, m) = r(k, m')$ , but this is not important.) Again, this means that the requirement will define functionals  $\Phi_{e,i}$  (for  $i < 2$ ) and expect to get traces  $\langle T_x^{e,i} \rangle_{x < \omega}$  for  $\Phi_{e,i}^{A_i}$  which obey a bound  $h_e$ , such that for all  $k \geq k^*(m)$ , the collection of  $x$  such that  $h_e(x) = k$  has size at least  $r(k, m)$ .

Then, given an order function  $g$ , we define an order function  $f = f_g$ , such that if c.e. sets  $A_0$  and  $A_1$  are jump-traceable via  $f$ , then  $A_0 \oplus A_1$  is jump-traceable via  $g$ . This is done in the following way. For each  $c < \omega$ , we partition  $\omega$  into intervals  $\langle I_k^c \rangle_{k \geq c}$  (so  $\min I_{k+1}^c = \max I_k^c + 1$ ), such that

$$|I_k^c| = \sum_{\{e: k^*(g(e)) \leq k\}} r(k, g(e))$$

and define a function  $f^c$  by letting  $f^c(x) = k$  if  $x \in I_k^c$ . Note that since  $\lim_e g(e) = \infty$ , for any  $k$ , for large enough  $e$  we have  $k^*(g(e)) > k$  and so the prescribed size of  $I_k^c$  is indeed finite. It is easy to see that  $f^c$  is an order function.

We also note that  $f^c(0) = c$ . By Lemma 1.5, there is an order function  $f$  such that for all  $x$  and  $c$ ,  $f(\alpha_c(x)) \leq f^c(x)$ . This is the required function.

Now given  $A_0$  and  $A_1$  which are jump-traceable via  $f$ , we get traces  $S^0, S^1$  for  $J^{A_0}, J^{A_1}$  which obey  $f$ . This allows us, uniformly in  $c$ , to get traces  $S^{c,0}, S^{c,1}$  for  $\Psi_c^{A_0}, \Psi_c^{A_1}$ , which obey  $f^c$ .

For each  $c$  and  $k \geq c$ , let

$$\langle N_{k,e}^c \rangle_{\{e: k^*(g(e)) \leq k\}}$$

be a partition of  $I_k^c$ , such that  $|N_{k,e}^c| = r(k, g(e))$ . For each  $c < \omega$ , we run the construction for all the  $e$  such that  $k^*(g(e)) \geq c$  simultaneously, with the  $e$ th requirement defining  $\Phi_{e,0}$  and  $\Phi_{e,1}$  with domain contained in  $\bigcup_{k \geq k^*(g(e))} N_{k,e}^c$  and using  $S^{c,0}$  and  $S^{c,1}$  as traces. Using Posner’s trick, we can effectively get an index  $c'$  such that for both  $i = 0, 1$ ,  $\Phi_i^{A_i} = \bigcup_{\{e: k^*(g(e)) \geq c\}} \Phi_{e,i}^{A_i} = \Psi_{c'}^{A_i}$ . By the recursion theorem, there is some  $c$  such that  $\Psi_c = \Psi_{c'}$  and so indeed  $T^i = S^{c,i}$  is a trace for  $\Phi_i^{A_i}$ , and so for large enough  $e$  (those  $e$  such that  $k^*(g(e)) \geq c$ ) we can get a trace  $T^{e,i}$  for  $\Phi_{e,i}^{A_i}$  which obeys  $h_e$ . For large enough  $e$ , this construction will trace  $J^{A_0 \oplus A_1}(e)$  with bound  $g(e)$ . Here end the global considerations; what is left to do is to fix  $e$  and  $m$ , define  $k^*(m)$  and  $r(k, m)$  (and so  $h_e$ ), and describe how, given traces for both  $\Phi_{e,i}^{A_i}$  which we define, we can trace  $J^{A_0 \oplus A_1}(e)$  with fewer than  $m$  mistakes.

*The local strategy*

So indeed, fix an  $e$  and an  $m$ . We define functionals  $\Phi_{e,i}$  and get traces  $\langle T_x^{e,i} \rangle$  for them, as described above, with bound  $h_e$  (which we soon define).

Let  $k^*(e) = \lfloor m/2 \rfloor$ . For any  $n$ , define a  $meta_0^n$ -box to be any singleton  $\{x\}$  and define a  $meta_{k+1}^n$ -box to be a collection of  $n + 2$  many  $meta_k^n$ -boxes. We often ignore the distinction between a  $meta_k^n$ -box  $M$  and  $\bigcup^{(k)} M$ , that is, the collection of numbers (inputs) which appear in  $meta_0^n$ -sub-boxes of  $M$ . In this sense, the size of a  $meta_k^n$ -box is  $(n + 2)^k$ . At the beginning, a meta-box  $M$  is an  $l$ -box (for either  $A_0$  or  $A_1$ ) if for all  $x \in M$ ,  $h_e(x) \leq l$ . At a later stage  $s$ , a meta-box  $M$  is an  $l$ -box for  $A_i$  if for all  $x \in M$ , we have  $h_e(x) - |T_x^{e,i}[s]| \leq l$ .

At the beginning, for all  $k \geq k^*(e)$  we wish to have two meta- $k_{k+1}$ -boxes which are  $k$ -boxes. We thus let  $r(k, m) = 2(k + 2)^{k+1}$ . Denote these two meta-boxes by  $N_k$  and  $N'_k$ . From now we drop all  $e$  subscripts, so  $\Phi_i = \Phi_{e,i}$ ,  $T_x^i = T_x^{e,i}$ ,  $h = h_e$ , etc.

At the beginning of a stage  $s$ , we have two numbers  $k_0^*[s]$  and  $k_1^*[s]$  (we start with  $k_i^*[0] = k^*(e)$ ). For  $i < 2$ , every  $k \in [k_i^*[s], s)$  has some priority  $p_i(k)[s] \in \frac{1}{2}\mathbb{N}$ . For such  $k$  we have finitely many meta- $k_{k+1}^{\lfloor p_i(k)[s] \rfloor}$ -boxes  $M_1^i(k), \dots, M_{d_i(k)}^i(k)[s]$ , each of which is *free* in the sense that for all  $x$  in any of these boxes, we have  $\Phi_i^{A_i}(x) \uparrow [s]$ .

First at stage  $s \geq k^*(e)$ , for both  $i = 0, 1$  we let  $M_1^i(s)[s], \dots, M_{s+2}^i(s)[s]$  be the meta- $s_s$ -sub-boxes of  $N_s$  (recall that these are all  $s$ -boxes). We let the priority  $p_i(s) = s$ .

Suppose now that we are given a computation  $J^{A_0 \oplus A_1}(e)[s]$  with use  $\sigma_0 \oplus \sigma_1$ , which we want to test. The test is done in steps, in increasing priority. We start with step  $s$ .

*Instructions for testing  $\sigma_0 \oplus \sigma_1$  at step  $n \in \frac{1}{2}\mathbb{N}$*

For  $i = 0, 1$ , if there is some  $k$  such that  $p_i(k)[s] = n$  (there will be at most one such  $k$  for each  $i$ ), then we take the last meta-box  $M = M_{d_i(k)}^i(k)[s]$ , and test  $\sigma_i$  on  $M$  by defining  $\Phi_i^{\sigma_i}(x) = \sigma_i$  for all  $x \in M$ . We then run the enumeration of the trace  $T^i$  and of  $A_i$  until one of the following happens:

- For all  $x \in M$ ,  $\sigma_i$  appears in  $T_x^i$  (we say that the test *returns*).
- $\sigma_i$  is not an initial segment of  $A_i$  anymore (we say that the test *fails*).

One of the two has to occur since  $T^i$  is indeed a trace for  $\Phi_i^{A_i}$ .

If all tests that were started (either none, one test for one  $\sigma_i$ , or two tests for both  $\sigma_i$ ) have returned, then we move to test at step  $n - 1/2$ ; but if  $n = 1$  then all tests at all levels have returned, and so we believe the computation  $J^{A_0 \oplus A_1}(e)[s]$  and trace it. In the latter case, from now we monitor this belief; we just keep defining  $p_i(s')$  and  $M_j^i(s')$  at later stages  $s'$ . If at a later stage  $t$  we discover that one of the  $\sigma_i$  was not in fact an initial segment of  $A_i$ , we update priorities as follows and go back to following the instructions above.

Also, if some test at step  $n$  fails, then we stop the testing at stage  $s$  and update priorities.

*Updating priorities*

Suppose that at some stage  $s$ , a test of  $\sigma_i$  at step  $n$  returns, but at a stage  $t \geq s$  we discover that  $\sigma_i \not\subset A_i$ . Let  $k$  be the level such that  $p_i(k)[s] = n$ . We do the following:

- (1) If  $k = k_i^*[s]$  then let  $k_i^*[t + 1] = k - 1$ .
- (2) Redefine  $p_i(k - 1)[t + 1] = n$  and  $d_i(k - 1)[t + 1] = \lfloor n \rfloor + 2$ , and let  $M_1^i(k - 1)[t + 1], \dots, M_{\lfloor n \rfloor + 2}^i(k - 1)[t + 1]$  be the collection of meta- $k_{k+1}^{\lfloor n \rfloor}$ -sub-boxes of  $M_{d_i(k)[s]}^i$  (which was the meta- $k_{k+1}^{\lfloor n \rfloor}$ -box used for the testing of  $\sigma_i$  at step  $n$  of stage  $s$ ).
- (3) If  $k = s$  then redefine  $p_i(k)[t + 1] = s + 1/2$ , redefine  $d_i(s)[t + 1] = s + 2$  and let  $M_1^i(s)[t + 1], \dots, M_{s+2}^i(s)[t + 1]$  be the meta- $s_s$ -sub-boxes of  $N'_s$  (note that these were untouched so far).

On the other side, if at stage  $t$  we still have  $\sigma_{1-i} \subset A_{1-i}[t]$ , and a test of  $\sigma_{1-i}$  at stage  $s$  at step  $n$  has started (and so returned), then we need to discard the meta-box  $M_{d_{1-i}(k)}^{1-i}(k)[s]$  (where again  $p_{1-i}(k)[s] = n$ ) and redefine  $d_{1-i}(k)[t + 1] = d_{1-i}(k)[s] - 1$ . We do this also if  $t = s$  and

the first test at step  $s$  has returned, but we immediately found out that  $\sigma_i \not\subset A_i$ , and the test on the  $\sigma_i$  side did not even return once.

*Justification and verification*

Let  $i < 2$ ,  $s \geq k^*(e)$ ,  $k \in [k_i^*[s], s)$ , and  $j \in \{1, \dots, d_i(k)[s]\}$ . Let  $n = p_i(k)[s]$ .

**Lemma 3.2.** *The meta $_{\perp}^{i,n}$ -box  $M_j^i(k)[s]$  is a  $k$ -box; indeed, for all  $x \in M_j^i(k)[s]$ , there are at least  $\perp n \perp - k$  many strings in  $T_x^i[s]$  which lie to the left of  $A_i[s]$  (and  $h(x) = \perp n \perp$ ).*

**Proof.** Let  $s$  be the least such that we define, for some level  $k$ ,  $p_i(k)[s] = n$ . Then  $k = \perp n \perp$  and there are two possibilities:

- If  $n \in \mathbb{N}$ , then  $s = n$ , the definition is made at the beginning of stage  $s$ , and we define  $M_1^i(s), \dots, M_{s+2}^i(s)[s]$  to be sub-boxes of  $N_s$ , which is an  $s$ -box.
- If  $n \notin \mathbb{N}$  then at stage  $s - 1$ , a test that began at stage  $\perp n \perp \leq s - 1$  (and returned on the  $\sigma_i$  side) is resolved by finding that  $\sigma_i \not\subset A_i[s]$ . We then define  $p_i(\perp n \perp)[s] = n$  and define  $M_1^i(\perp n \perp), \dots, M_{\perp n \perp + 2}^i(\perp n \perp)[s]$  to be sub-boxes of  $N'_{\perp n \perp}$ , which is an  $\perp n \perp$ -box.

In either case, the  $M_j^i(\perp n \perp)[s]$  are  $\perp n \perp$ -boxes, so indeed for each  $x$  in these meta-boxes,  $h(x) = \perp n \perp$ , and  $T_x^i$  indeed contains at least  $\perp n \perp - \perp n \perp$  many strings.

By induction, if  $n = p_i(k)[t]$  at a later stage  $t$ , then for all  $j$ ,  $M_j^i(k)[t]$  is a sub-box of some  $M_{j'}^i(\perp n \perp)[s]$ , and so for all  $x \in M_j^i(k)[t]$  we have  $h(x) = \perp n \perp$ .

Suppose that at stage  $t$  we redefine  $p_i(k - 1)[t + 1] = n$  and redefine  $M_j^i(k - 1)[t + 1]$ . Then at some stage  $r \leq t$  we defined, for all  $x \in M_{d_i(k)}^i[s]$ ,  $\Phi_i^{\sigma_i}(x) = \sigma_i$  where  $\sigma_i \subset A_i[r]$  but  $\sigma_i \not\subset A_i[t + 1]$ . By induction, at stage  $r$  there are at least  $\perp n \perp - k$  many strings in  $T_x^i[r]$  that lie to the left of  $A_i[r]$ ; they all must be distinct from  $\sigma_i$ . The test at stage  $r$  returned, which means that  $\sigma_i \in T_x^i[r + 1]$ ; thus  $T_x^i[t + 1]$  contains at least  $\perp n \perp - (k - 1)$  many strings that lie to the left of  $A_i[t + 1]$ .  $\square$

**Lemma 3.3.** *The sequence  $k_i^*[s]$  is non-increasing with  $s$ ; for all  $s$  we have  $k_i^*[s] > 0$ .*

**Proof.** By Lemma 3.2, for all  $j \leq d_i(k_i^*[s])[s]$  and  $x \in M_j^i(k_i^*[s])[s]$  we have  $|T_x^i| \geq h(x) - k_i^*[s]$ ; as  $|T_x^i| < h(x)$  we must have  $k_i^*[s] > 0$ .  $\square$

**Lemma 3.4.** *The sequence  $p_i(k)[s]$  is strictly increasing with  $k$ .*

**Proof.** Assume this at the beginning of stage  $t$ . We first define  $p_i(t)[t] = t$ ; all numbers used prior to this stage were below  $t$ .

Now suppose that at stage  $t$  we update priorities because of a test which returns at some stage  $s \leq t$  is found to be incorrect. The induction hypothesis for  $s$ , and the instructions for testing, ensure that the collection of levels  $k$  for which a  $\sigma_i$ -test has returned at stage  $s$  is an interval  $[k_0, s]$ . Priorities then shift one step downward to the interval  $[k_0 - 1, s - 1]$ ; the sequence of priorities is still increasing. Finally, a new priority  $s + 1/2$  is given to level  $s$ ; it is greater than the priorities for levels  $k < s$  (which get priority at most  $s$ ) but smaller than the priority  $k$  which is given to all levels  $k \in (s, t]$ .  $\square$

Also note that we always have  $p_i(k) \geq k$  because we start with  $p_i(k)[k] = k$ , then perhaps later change it to  $k + 1/2$ , and from then on it never decreases.

The following key calculation ensures that we never run out of boxes at any level, on either side, so the construction can go on and never get stuck. It ties losses of boxes on one side to gains on the other. For any  $k \in [k_i^*[s], s]$ , let  $l_i(k)[s]$  be the least level  $l$  such that  $p_{1-i}(l) \geq p_i(k)[s]$ . Such a level must exist because at the beginning of the stage we let  $p_{1-i}(s) = s$ , which is greater or equal to  $p_i(k)[s]$  for any  $k \leq s$ . Thus  $l_i(k)[s] \geq 1$ .

**Lemma 3.5.** *At stage  $s$ , for  $i < 2$  and  $k \in [k_i^*[s], s]$ , the number  $d_i(k)[s]$  of meta- $k$ -boxes is at least:*

- $l_i(k)[s]$ , if  $p_{1-i}(l_i(k)) > p_i(k)[s]$ ;
- $l_i(k)[s] + 1$ , if  $p_{1-i}(l_i(k)) = p_i(k)[s]$ .

**Proof.** This goes by induction on the stage. Suppose this is true at the end of stage  $t - 1$ ; we consider what changes we may have at stage  $t$ .

First at stage  $t$ , we define  $p_i(t) = t = p_{1-i}(t)$ . We thus have  $l_i(t) = t$  and  $p_{1-i}(l_i(t)) = p_i(t)$  and so we are required to have  $t + 1$  many  $t$ -boxes; we actually have  $d_i(t)[t] = t + 2$  many.

Suppose that a test which began at stage  $s \leq t$  is resolved at stage  $t$ , and priorities are updated.

There are two sides. Suppose first that  $\sigma_i \not\subset A_i[t + 1]$ , and that  $d_i(k)[t + 1] \neq d_i(k)[t]$ . If  $k < s$ , then a test at level  $k + 1$  returned at stage  $s$ . We then redefine  $d_i(k)[t + 1] = \lfloor n \rfloor + 2$  where  $n = p_i(k)[t + 1]$  ( $= p_i(k + 1)[s]$ ). As mentioned, we always have  $p_{1-i}(\lceil n \rceil) \geq \lceil n \rceil$  and so  $l_i(k)[t + 1] \leq \lfloor n \rfloor + 1$ , so we are in the clear. If, however,  $k = s$ , then we redefine  $d_i(s)[t + 1] = s + 2$  and  $p_i(s)[t + 1] = s + 1/2$ ; again,  $p_{1-i}(s + 1)[t + 1] \geq s + 1$  and so  $l_i(k)[t + 1] \leq s + 1$ , so  $d_i(s) \geq l_i(k) + 1[t + 1]$  as required.

Now take the losing side: suppose that  $\sigma_i \subset A_i[t + 1]$ . We may have lost some meta-boxes on this side; but changing priorities on the other side give us compensation. Let  $k \in [k_i^*[t], t]$ ; before anything else, we note that if  $k > s$  then  $d_i(k)[t + 1] = k + 2$ ,  $l_i(k)[t + 1] = k$  and  $p_{1-i}(k)[t + 1] = k$ , so there are sufficiently many  $k$ -boxes. We assume then that  $k \leq s$ .

We also examine the case that  $k = s$ . In this case,  $d_i(k)[t + 1] = d_i(k)[s] - 1 = s + 1$ . We have  $p_i(k)[t + 1] = s$  and  $l_i(k)[t + 1] \leq s$  and so  $d_i(k) \geq l_i(k) + 1[t + 1]$ . We assume from now that  $k < s$ .

Let  $n = p_i(k)[t + 1] = p_i(k)[s]$ . We note that if there is no  $k'$  such that  $p_{1-i}(k')[s] = n$ , then there is no  $k'$  such that  $p_{1-i}(k')[t + 1] = n$ . This is because the only priority we may add at stage  $t$  (after the initial part of the stage) is  $s + 1/2$ , and  $n < s$ . Thus, if  $n' = p_{1-i}(l_i(k))[s] > n$  then  $p_{1-i}(l_i(k))[s] \geq n' > n$ , because there are three possibilities for the behaviour of  $l_i(k)$  and  $p_{1-i}(l_i(k))$ . Let  $k' = l_i(k)[s]$ , and note that  $k' < s$ .

- (1) A test for  $\sigma_{1-i}$  at step  $n'$  of stage  $s$  returns. In this case,  $l_i(k)[t + 1] = k' - 1$  and  $p_{1-i}(l_i(k))[t + 1] = n'$ .
- (2) A test for  $\sigma_{1-i}$  at level  $k'$  (at stage  $s$ ) does not return, but a test for  $\sigma_{1-i}$  at level  $k' + 1$  does return. In this case the priority  $n'$  is removed on side  $1 - i$  at stage  $t$ ; we redefine  $p_{1-i}(k')[t + 1] = p_{1-i}(k' + 1)[s] > n'$ . However, we still have  $l_i(k)[t + 1] = k'$  because (if  $k' > k_{1-i}^*[s]$ ) we still have  $p_{1-i}(k' - 1)[t + 1] = p_{1-i}(k' - 1)[s] < n$ .
- (3) A test for  $\sigma_{1-i}$  at level  $k' + 1$  is not started or does not return. In this case there is no change at level  $k'$  and  $k' - 1$ ; we have  $l_i(k)[t + 1] = k'$  and  $p_{1-i}(k')[t + 1] = n'$ .

In any case, we see that we cannot have a case at which  $l_i(k)$  increases from stage  $s$  to stage  $t + 1$ , or that  $p_{1-i}(l_i(k))[s] > n$  but  $p_{1-i}(l_i(k))[t + 1] = n$ . Thus the required number of  $k$ -meta-boxes does not increase from stage  $s$  to stage  $t + 1$ . Thus we need only to check what happens if  $d_i(k)[t + 1] = d_i(k)[s] - 1$ . Assume this is the case; we check each of the three scenarios above.

In case (1), the number of required boxes has decreased by one; this exactly compensates the loss. Case (3) is not possible if a  $k$ -box is lost; this is because a test at step  $n$  is started only after a test for  $\sigma_{1-i}$  at step  $p_{1-i}(k' + 1)[s]$  has returned.

The same argument shows that if case (2) holds and we lost a  $k$ -box, then necessarily  $n' = n$ . But then  $d_i(k)[s] \geq k' + 1$ , but the fact that now  $p_{1-i}(k')[t + 1] > n$  implies that the number of required boxes has just decreased by one, to  $k'$ ; again the loss is compensated.  $\square$

We are now ready to finish. We note that if indeed  $J^{A_0 \oplus A_1}(e)$  converges, then at some point the correct computation appears and is tested. Of course all tests must return, and so the correct value will be traced.

If, on the other hand, a value  $J^{A_0 \oplus A_1}(e)[s]$  is traced at stage  $s$  because all tests return, but at a later stage  $t$  we discover that this computation is incorrect, say  $\sigma_i \notin A_i[t + 1]$ , then  $k_i^*[t + 1] < k_i^*[t]$ . As we always have  $k_i^*[r] \geq 1$ , this must happen fewer than  $2k^*(e) \leq m$  many times. It follows that the total number of values traced is at most  $m$ , as required.

#### 4. Strongly jump-traceable c.e. sets are $K$ -trivial

Let  $A$  be strongly jump-traceable; we prove that it is low for  $K$ , and hence  $K$ -trivial. We need to cover  $U^A$  by an oracle-free machine, obtained via the Kraft–Chaitin theorem. We enumerate  $A$  and thus approximate  $U^A$ . When a string  $\sigma$  enters the domain of  $U^A$  we need to decide whether we believe the  $A$ -computation that put  $\sigma$  in  $\text{dom } U^A$ ; again the idea is to test this by testing the use  $\rho \subset A[s]$  which enumerated  $\sigma$  into  $\text{dom } U^A[s]$ ; again the naive idea is to pick some input  $x$  and define a functional  $\Psi^\rho(x) = \rho$ . Then  $\Psi^A$  is traced by a trace  $\langle T_x \rangle$ ; only if  $\rho$  is traced do we believe it is indeed an initial segment of  $A$  and so believe that  $U^A(\sigma)$  is a correct computation. We can then enumerate  $(|\sigma|, U^A(\sigma))$  into a Kraft–Chaitin set we build and so ensure that  $K(U^A(\sigma)) \leq^+ |\sigma|$ .

The combinatorics of the construction aim to ensure that we indeed build a Kraft–Chaitin set; that is, the total amount of mass that we believe at some stage of the construction is finite. This would of course be ensured if we only believed correct computations, as  $\mu(\text{dom } U^A)$  is finite. However, the size of most  $T_x$  is greater than 1, and so an incorrect  $\rho$  may be believed. We need to limit the mass of the errors.

To handle this calculation, rather than treat each string  $\sigma$  individually, we batch strings up in pieces of mass. When we have a collection of strings in  $\text{dom } U^A$  whose total mass is  $2^{-k}$  we verify  $A$  up to a use that puts them all in  $\text{dom } U^A$ . The greater  $2^{-k}$  is, the more stringent the test will be (ideally, in the sense that the size of  $T_x$  is smaller). We will put a limit  $m_k$  on the amount of times that a piece of size  $2^{-k}$  can be believed and yet be incorrect. The argument will succeed if

$$\sum_{k < \omega} m_k 2^{-k}$$

is finite.

Once we use an input  $x$  to verify an  $A$ -correct piece, it cannot be used again for any testing, as  $\Psi^A(x)$  becomes defined permanently. Following the naive strategy, we would need at least  $2^k$

many inputs for testing pieces of size  $2^{-k}$ . Even a single error on each  $x$  (and there will be more, as the size of  $T_x$  has to go to infinity) means that  $m_k \geq 2^k$  is too large. Again, the rest of the construction is a combinatorial strategy: which inputs are assigned to which pieces in such a way as to ensure that the number of possible errors  $m_k$  is sufficiently small. The strategy has two ingredients.

First, we note that two pieces of size  $2^{-k}$  can be combined into a single piece of size  $2^{-(k-1)}$ . So if we are testing one such piece, and another piece, with comparable use, appears, then we can let the testing machinery for  $2^{-(k-1)}$  take over. Thus, even though we need several testing locations for  $2^{-k}$  (for example if a third comparable piece appears), at any stage, the testing at  $2^{-k}$  is really responsible for at most one such piece.

The naive reader would imagine that it is now sufficient to let the size of  $T_x$  (for  $x$  testing  $2^{-k}$ -pieces) be something like  $k$  and be done. However, the opponent's spoiling strategy would be to "drip-feed" small mass that aggregates to larger pieces only slowly (this is similar to the situation in decanter constructions). In particular, fixing some small  $2^{-k}$ , the opponent will first give us  $k$  pieces (of incomparable use) one after the other (so as to change  $A$  and remove one before giving us a new one). At each such occurrence we would need to use the input  $x$  devoted to the first  $2^{-k}$  piece, because at each such stage we only see one. Once the amount of errors we get from using  $x$  for testing is filled ( $T_x$  fills up to the maximum allowed size) the opponent gives us one correct piece of size  $2^{-(k-1)}$  and then moves on to gives us  $k$  more incorrect pieces which we test on the next  $x$ . Overall, we get  $k$  errors on *each*  $x$  used for  $2^{-k}$ -pieces. As we already agreed that we need something like  $2^k$  many such  $x$ 's, we are back in trouble.

Every error helps us make progress as the opponent has to give up one possible value in some  $T_x$ ; fewer possible mistakes on  $x$  are allowed in the future. The solution is to make every single error count in our favour in all future testings of pieces of size  $2^{-k}$ . In other words, what we need to do is to maximize the benefit that is given by a single mistake; we make sure that a single mistake on *some* piece will mean one less possible mistake on *every* other piece. In other words, we again use meta-boxes.

In the beginning, rather than just testing a piece on a single input  $x$ , we test it simultaneously on a large set of inputs and only believe it is correct if the use shows up in the trace of every input tested. If this is believed and more pieces show up then we use them on other large sets of inputs. If, however, one of these is incorrect, then we later have a large collection of inputs  $x$  for which the number of possible errors is reduced. We can then break up this collection into  $2^k$  many smaller collections and keep working only with such  $x$ 's.

This can be geometrically visualised as follows. If the naive strategy was played on a sequence of inputs  $x$ , we now have an  $m_k$ -dimensional cube of inputs, each side of which has length  $2^k$ . In the beginning we test each piece on one hyperplane. If the testing on some hyperplane is believed and later found to be incorrect then from then on we work in that hyperplane, which becomes the new cube for testing pieces of size  $2^{-k}$ ; we test on hyperplanes of the new cube. If the size of  $T_x$  for each  $x$  in the cube is at most  $m_k$  then we never "run out of dimensions."

#### 4.1. The formal construction and proof of Theorem 1.1

Given  $c < \omega$  (say  $c > 1$ ), we partition  $\omega$  into intervals  $\langle M_k^c \rangle_{k < \omega}$  such that  $|M_k^c| = 2^{k(k+c)}$ . For  $x \in M_k^c$  we let  $h_c(x) = k + c - 1$ . By Lemma 1.5, we get an order function  $\tilde{h}$  such that for all  $c$  and  $x$ ,  $\tilde{h}(\alpha_c(x)) \leq h_c(x)$ . We fix a trace for  $J^A$  with bound  $\tilde{h}$ . From this trace, we can, uniformly in  $c$ , get a trace for  $\Psi_c^A$  with bound  $h_c$ .

Note that  $h_c$  grows roughly like  $\sqrt{\log x}$ . This gives us the bound mentioned in the introduction. The exact bound for  $h$  may be slower, depending on the way each  $\Psi_c$  is coded into the jump function  $J$ .

In our construction, we define a functional  $\Psi$ ; by the recursion theorem we know some  $c$  such that for all  $X \in 2^\omega$ ,  $\Psi^X = \Psi_c^X$ . We let  $M_k[0] = M_k^c$  and let  $\langle T_x \rangle$  be the trace for  $\Psi^A$  with bound  $h = h_c$ .

*Usage of  $\Psi$*

Again, the axioms that we enumerate into  $\Psi$  are all of the form  $\Psi^\rho(x) = \rho$  for some  $\rho \in 2^{<\omega}$  and  $x < \omega$ . We only enumerate such an axiom at stage  $s$  if  $\rho \subset A[s]$ .

Let  $R_k = \{m2^{-k} : m = 0, 1, 2, \dots, 2^k\}$ , and let  $R_k^+ = R_k \setminus \{0\}$ .

*The boxes*

We can label the elements of  $M_k[0]$  so that

$$M_k[0] = \{x_f : f : (k + c) \rightarrow R_k^+\}.$$

[So  $M_k[0]$  is a  $(k + c)$ -dimensional cube; the length of each side is  $2^k$ .]

At stage  $s$ , for each  $k$  we have a function  $g_k[s] : d_k[s] \rightarrow R_k^+$  (where  $d_k[s] < k + c$ ) which determines the current value of  $M_k$ :

$$M_k[s] = \{x_f \in M_k[0] : g_k[s] \subset f\}$$

(so  $d_k[0] = 0$  and  $g_k[s]$  is the empty function). Thus  $M_k[s]$  is a  $(k + c - d_k)$ -dimensional cube.

For  $q \in R_k^+$ , we let

$$N_k(q)[s] = \{x_f \in M_k[s] : f(d_k[s]) = q\};$$

this is the  $(2^k \cdot q)$ th hyper-plane of  $M_k[s]$ .

*Strings*

Recall that for any string  $\rho \in 2^{<\omega}$ , we let  $\Omega^\rho$  be the measure of the domain of  $U^\rho$ , the universal machine with oracle  $\rho$ . Note that  $\rho \mapsto \Omega^\rho$  is monotone: if  $\rho \subset \nu$  then  $\Omega^\rho \leq \Omega^\nu$ . We assume that the running time of any computation with oracle  $\rho$  is at most  $|\rho|$  steps, and so:

- The maps  $\rho \mapsto U^\rho$  and so  $\rho \mapsto \Omega^\rho$  are computable.
- For all  $\sigma \in \text{dom } U^\rho$ ,  $|\sigma| \leq |\rho|$ .

It follows that  $\Omega^\rho$  is a multiple of  $2^{-|\rho|}$ , in other words, is an element of  $R_{|\rho|}$ . Also note that since  $\langle \rangle \notin U^X$  for any  $X$ , the assumption implies that  $U^\langle \rangle$  is empty and so  $\Omega^\langle \rangle = 0$ .

Let  $q$  be any rational. For any  $\nu \in 2^{<\omega}$  such that  $\Omega^\nu \geq q$ , we let  $q^\nu(q)$  be the shortest string  $\rho \subseteq \nu$  such that  $\Omega^\rho \geq q$ . This operation is monotone with  $q$ : if  $q < q'$  and  $\Omega^\nu \geq q'$  then  $q^\nu(q) \subseteq q^\nu(q')$ .

*The standard configuration*

At the beginning of stage  $s$  of the construction, we are given  $A$  at some point of its enumeration, which we denote by  $A[s]$  (more than one number may go into  $A$  at each stage, as we describe below).

At the beginning of the stage, the cubes  $\langle M_k \rangle$  will be in the *standard configuration* for the stage. Fix  $k \leq s$  and  $q \in R_k^+$ .

- If  $q \leq \Omega^{A[s] \uparrow s}$  then for all  $x \in N_k(q)[s]$  we have  $\Psi^\rho(x) \downarrow = \rho[s]$ , where  $\rho = \varrho^{A[s] \uparrow s}(q)$ .
- If  $q > \Omega^{A[s] \uparrow s}$  then for all  $x \in N_k(q)[s]$ , we have  $\Psi^{A[s]}(x) \uparrow [s]$ .

Further, for all  $k > s$  and all  $x \in M_k[s]$ , no definition of  $\Psi(x)$  (for any oracle) was ever made.

Suppose that  $\rho \subseteq A[s] \uparrow s$ . We say that  $\rho$  is *semi-confirmed* at some point during stage  $s$  if for all  $x$  such that  $\Psi^\rho(x) \downarrow = \rho$  at stage  $s$ , we have  $\rho \in T_x$  at that given point (which may be the beginning of the stage or later). We say that  $\rho$  is *confirmed* if every  $\rho' \subseteq \rho$  is semi-confirmed.

Note that the empty string is (emptily) confirmed at every stage. This is because for no  $x$  do we ever define  $\Psi^\langle \rangle(x) \downarrow = \langle \rangle$ ; this is because  $\Omega^\langle \rangle = 0$  and so for no  $s$  and no  $q > 0$  do we have  $\langle \rangle = \varrho^{A[s] \uparrow s}(q)$ .

*Construction*

At stage  $s$ , do the following:

- (1) Speed up the enumeration of  $A$  and of  $\langle T_x \rangle$  (to get  $A[s + 1]$  and  $T_x[s + 1]$ ) so that for all  $\rho \subseteq A[s] \uparrow s$ , one of the following holds:

- (a)  $\rho$  is confirmed.
- (b)  $\rho$  is not an initial segment of  $A$  anymore.

One of the two must happen because  $\langle T_x \rangle$  traces  $\Psi^A$ .

- (2) For any  $k \leq s$ , look for some  $q \in R_k^+$  such that  $q \leq \Omega^{A[s] \uparrow s}$  and such that for  $\rho = \varrho^{A[s] \uparrow s}(q)$  we have:

- $\rho$  was confirmed at the beginning of the stage; but
- $\rho \not\subseteq A[s + 1]$ .

If there is such a  $q$ , pick one, and extend  $g_k$  by setting  $g_k(d_k) = q$ . Thus  $d_k[s + 1] = d_k[s] + 1$  and  $M_k[s + 1] = N_k(q)[s]$ .

- (3) Next, define  $\Psi$  as necessary so that the standard configuration will hold at the beginning of stage  $s + 1$ .

*Justification*

We need to explain why the construction never gets stuck. There are two issues:

1. Why do we not “run out of dimensions”? That is, why can we always increase  $d_k$  if we are asked to?
2. Why can we always return to the next standard configuration?

For the first, we prove the following.

**Lemma 4.1.** *For every  $x \in M_k[s]$ , there are at least  $d_k[s]$  many strings  $\rho \in T_x[s]$  which lie (lexicographically) to the left of  $A[s]$ .*

**Proof.** Suppose that during stage  $s$ , we increase  $d_k$  by one. This is witnessed by some  $q \in R_k^+$  and a string  $\rho = \varrho^{A[s] \uparrow s}(q)$  which was confirmed at the beginning of the stage; we set  $M_k[s + 1] = N_k(q)[s]$ . The confirmation implies that for all  $x \in N_k(q)[s]$ ,  $\rho \in T_x$ . But we also know that  $\rho \subseteq A[s]$  and  $\rho \not\subseteq A[s + 1]$ . As  $A$  is c.e., it had to move to the right of  $\rho$ . If we increase

$d_k$  at stages  $s_1 < s_2$  (witnessed by strings  $\rho_1$  and  $\rho_2$ ) then  $\rho_1$  lies to the left of  $A[s_1 + 1]$  whereas  $\rho_2$  is an initial segment of  $A[s_2]$  (which is not left of  $A[s_1 + 1]$ ). Thus  $\rho_1$  lies to the left of  $\rho_2$ , and in particular, they are distinct.  $\square$

Since for all  $x \in M_k[0]$ ,  $h(x) = k + c - 1$ , we know that for all such  $x$ ,  $|T_x| \leq k + c - 1$ , which implies that for all  $s$  we must have  $d_k[s] < k + c$ .

For the second issue, let  $k < \omega$ .

If  $M_k[s + 1] \neq M_k[s]$ , witnessed by some  $q \in R_k^+$  and by  $\rho = \varrho^{A[s] \upharpoonright s}(q)$ , then for all  $x \in M_k[s + 1]$  we know that  $\Psi^\rho(x) \downarrow = \rho$ ; so for no proper initial segment  $\rho' \subsetneq \rho$  do we have  $\Psi^{\rho'}(x) \downarrow [s]$ . As  $\rho$  is not an initial segment of  $A[s + 1]$  we must have  $\Psi^{A[s+1] \upharpoonright s}(x) \uparrow$  so we are free to make any definitions we like (recall that no definitions to right of  $A[s]$  are made before stage  $s$ ).

For  $k = s + 1$ , we know that  $M_k$  was empty up to stage  $s$ , so we have a clean slate there.

Suppose that  $k \leq s$  and that  $M_k[s + 1] = M_k[s]$ . Let  $q \in R_k^+$  such that  $q \leq \Omega^{A[s+1] \upharpoonright s+1}$ , and let  $x \in N_k(q)[s + 1]$  ( $= N_k(q)[s]$ ). We want to define  $\Psi^\rho(x) \downarrow = \rho$  where  $\rho = \varrho^{A[s+1] \upharpoonright s+1}(q)$ .

If  $\rho \not\subset A[s]$  then  $\rho$  lies to the right of  $A[s]$ , and so  $\Psi^\rho(x) \uparrow$  for all  $x \in M_k[s]$ . Suppose that  $\rho \subset A[s]$ . There are two possibilities:

- (1) If  $|\rho| \leq s$  then  $\rho = \varrho^{A[s] \upharpoonright s}(q)$  and so we already have  $\Psi^\rho(x) \downarrow = \rho$  for all  $x \in N_k(q)[s]$ .
- (2) If  $|\rho| = s + 1$  then (since we know that for every proper initial segment  $\rho'$  of  $\rho$  we have  $q > \Omega^{\rho'}$ ) we have  $q > \Omega^{A[s] \upharpoonright s}$ . Since the standard configuration held at the beginning of stage  $s$ , we have  $\Psi^{A[s] \upharpoonright s}(x) \uparrow$  at the beginning of the stage (for all  $x \in N_k(q)$ ). Thus we are free to define  $\Psi^\rho(x)$  as we wish.

This concludes the justifications.

*Verification*

Let  $s$  be a stage. We let  $\rho^*[s]$  be the longest string (of length at most  $s$ ) which is a common initial segment of both  $A[s]$  and  $A[s + 1]$ . Thus  $\rho^*[s]$  is the longest string which is confirmed at the beginning of stage  $s + 1$ .

We define

$$L = \bigcup \{U^{\rho^*[s]} : s < \omega\} = \{(\sigma, \tau) : \exists s U^{\rho^*[s]}(\sigma) = \tau\}.$$

This is a c.e. set.

**Lemma 4.2.**  $U^A \subseteq L$ .

**Proof.** Suppose that  $U^A(\sigma) = \tau$ . Let  $\rho \subset A$  some string such that  $U^\rho(\sigma) = \tau$ . Let  $s > |\rho|$  be late enough so that  $\rho \subset A[s], A[s + 1]$ . Then  $\rho \subseteq \rho^*[s]$  and so  $(\sigma, \tau) \in L$ .  $\square$

The remainder of the verification is devoted to prove the following:

**Lemma 4.3.**

$$\sum_{(\sigma, \tau) \in L} 2^{-|\sigma|}$$

is finite.

This would show that

$$\{(|\sigma|, \tau) : (\sigma, \tau) \in L\}$$

is a Kraft–Chaitin set and so there is some constant  $e$  such that for all  $(\sigma, \tau) \in L$ ,  $K(\tau) \leq |\sigma| + e$ . Together with Lemma 4.2, we see that  $A$  is low for  $K$ : for all  $\tau$ ,  $K(\tau) \leq K^A(\tau) + e$ .

Now  $L$  has two parts:  $U^A$  and  $L \setminus U^A$ . We know of course that  $\mu(\text{dom } U^A)$  is finite, and so we need to show that

$$\sum_{(\sigma, \tau) \in L \setminus U^A} 2^{-|\sigma|}$$

is finite.

Let  $s$  be a stage. For  $k \leq s$ , let  $q_k[s]$  be the greatest element of  $R_k$  not greater than  $\Omega^{\rho^*[s]}$ . This is monotone: if  $k < k' \leq s$  then  $q_k[s] \leq q_{k'}[s]$  because  $R_k \subset R_{k'}$ . Note that  $|\rho^*[s]| \leq s$  and so  $\Omega^{\rho^*[s]}$  is an integer multiple of  $2^{-s}$ ; it follows that  $q_s[s] = \Omega^{\rho^*[s]}$ . Also, since for all  $\rho$  we have  $\Omega^\rho < 1$ , we must have  $q_0[s] = 0$ .

Let  $v_k[s] = \Omega^{\rho^*[s]}(q_k[s])$ . By the monotonicity just mentioned, if  $k < k' \leq s$  then  $v_k[s] \subseteq v_{k'}[s]$  and so  $\Omega^{v_k[s]} \leq \Omega^{v_{k'}[s]}$ . Also,  $v_0[s] = \langle \rangle$ , and  $\Omega^{v_s[s]} = \Omega^{\rho^*[s]}$  (so  $U^{v_s[s]} = U^{\rho^*[s]}$ ).

The following is the key calculation.

**Lemma 4.4.** For all  $k \in \{1, 2, \dots, s\}$ ,

$$\Omega^{v_k[s]} - \Omega^{v_{k-1}[s]} \leq 2 \cdot 2^{-k}.$$

**Proof.** We know that  $q_{k-1}[s] \leq \Omega^{v_{k-1}[s]}$  and that  $\Omega^{v_{k-1}[s]} \leq \Omega^{v_k[s]}$ . On the other hand,  $\Omega^{v_k[s]} \leq \Omega^{\rho^*[s]}$  and  $\Omega^{\rho^*[s]} \leq q_{k-1}[s] + 2^{-(k-1)}$ . So overall,

$$q_{k-1}[s] \leq \Omega^{v_{k-1}[s]} \leq \Omega^{v_k[s]} \leq q_{k-1}[s] + 2 \cdot 2^{-k}. \quad \square$$

If  $(\sigma, \tau) \in L \setminus U^A$ , then we will find some  $k < \omega$  and some stage  $t$  and “charge” the mistake of adding  $(\sigma, \tau)$  to  $L$  against  $k$  at stage  $t$ ; we denote the collection of charged mass by  $L_{k,t}$ . Formally, we will define sets  $L_{k,t}$  and show that:

(1) For each  $k$  and  $t$ , the mass of  $L_{k,t}$ , namely

$$\sum_{(\sigma, \tau) \in L_{k,t}} 2^{-|\sigma|},$$

is at most  $2 \cdot 2^{-k}$ .

(2) 
$$L \setminus U^A \subseteq \bigcup_{k,t} L_{k,t}.$$

(3) For each  $k$ , there are at most  $k + c$  many stages  $t$  such that  $L_{k,t}$  is non-empty.

Given these facts, we get that

$$\sum_{(\sigma, \tau) \in L \setminus U^A} 2^{-|\sigma|} \leq \sum_{k,t} \sum_{(\sigma, \tau) \in L_{k,t}} 2^{-|\sigma|} \leq \sum_k 2(k+c)2^{-k}$$

which is finite as required. We turn to define  $L_{k,t}$  and to prove (1)–(3).

Fix  $t$  and  $k$  such that  $1 \leq k \leq t$ . If  $v_k[t] \not\subset A[t+2]$  then we let

$$L_{k,t} = U^{v_k[t]} \setminus U^{v_{k-1}[t]}.$$

Otherwise, we let  $L_{k,t} = \emptyset$ .

Fact (1) follows from Lemma 4.4:

$$\sum_{(\sigma, \tau) \in L_{k,t}} 2^{-|\sigma|} = \mu(\text{dom}(U^{v_k[t]} \setminus U^{v_{k-1}[t]})) = \Omega^{v_k[t]} - \Omega^{v_{k-1}[t]} \leq 2 \cdot 2^{-k}.$$

**Lemma 4.5.**

$$L \setminus U^A \subseteq \bigcup_{k,t} L_{k,t}.$$

**Proof.** Let  $(\sigma, \tau) \in L \setminus U^A$ .

Let  $\rho$  be the shortest string such that  $(\sigma, \tau) \in U^\rho$  and for some  $s, \rho \subset \rho^*[s]$ . Find such a stage  $s$  (so  $\rho \subset A[s], A[s+1]$ ). Since  $\rho \not\subset A$ , there is a stage  $t \geq s$  such that  $\rho \subset A[t], A[t+1]$  but  $\rho \not\subset A[t+2]$ .

Since  $\rho \subset \rho^*[t]$  and  $U^{\rho^*[t]} = U^{v_t[t]}$ , by minimality of  $\rho$ , we have  $\rho \subseteq v_t[t]$ . Since  $v_0[t] = \langle \rangle$ , there is some  $k \in [1, t]$  such that  $v_{k-1}[t] \subsetneq \rho \subseteq v_k[t]$ .

Since  $\rho \subseteq v_k[t]$ , we have  $(\sigma, \tau) \in U^{v_k[t]}$ . Since  $v_{k-1}[t] \subset \rho^*[t]$ , the minimality of  $\rho$  implies that  $(\sigma, \tau) \notin U^{v_{k-1}[t]}$ . Finally,  $\rho \not\subset A[t+2]$  and so  $v_k[t] \not\subset A[t+2]$ . Thus  $(\sigma, \tau) \in L_{k,t}$ .  $\square$

Finally, we prove fact (3) by showing the following:

**Lemma 4.6.** *Suppose that  $L_{k,t} \neq \emptyset$ . Then  $M_k[t+1] \neq M_k[t+2]$ .*

**Proof.** Suppose that  $L_{k,t} \neq \emptyset$ , so  $v_k[t] \not\subset A[t+2]$ . Let  $q = q_k[t]$ . Then  $v_k[t] = \varrho^{\rho^*[t]}(q) = \varrho^{A[t]t}(q)$ . Since  $v_k[t] \subseteq \rho^*[t]$ , it was confirmed at the beginning of stage  $t+1$ . Also,  $q > 0$  because otherwise  $v_k[t] = \langle \rangle$  and then  $U^{v_k[t]}$ , and so  $L_{k,t}$ , would be empty.

But then all the conditions for redefining  $M_k$  during stage  $t+1$  are fulfilled.  $\square$

### 5. Strongly jump-traceable c.e. sets do not ML-cup

The framework from the previous section can be adapted to provide a proof of Theorem 1.3, that no strongly jump-traceable, c.e. set  $A$  can be joined above  $\mathbf{0}'$  by an incomplete Martin–Löf random set. To show this (following in part Nies’s construction [22] of a set that does not ML-cup), we take a set  $Y$  of degree  $\mathbf{0}'$  and are given some Turing functional  $\Gamma$ ; we want to construct a

Solovay test which contains all incomplete reals  $X$  such that  $\Gamma(A \oplus X) = Y$ .<sup>5</sup> To assist with that, we build our own Turing functional  $\Delta$  and ensure that if  $\Gamma(A \oplus X) = Y$  then either  $\Delta(X) = Y$  or we can cover  $X$  in our Solovay test.

Again the idea is to use traceability to certify given computations, this time of the form  $\Gamma(A \oplus \sigma) \subset Y[s]$ . Once such a computation is certified, we will declare that  $\Delta(\sigma)$  computes that initial segment  $\tau$  of  $Y$  which was given by  $\Gamma(A \oplus \sigma)$ . Three conditions must hold in order for us to be worried by such a declaration:

- (1)  $A$  changes (below the use of the  $\Gamma$  computation);
- (2)  $Y$  changes (so that  $\tau \not\subset Y$ );
- (3) a new computation  $\Gamma(A \oplus \sigma) \subset Y$  appears (with the new versions of  $A$  and  $Y$ ).

In this case we would like to declare that  $\Delta(\sigma) = \tau'$ , the new initial segment of  $Y$ ; but  $\tau$  and  $\tau'$  are incompatible and this would make  $\Delta$  inconsistent. Note that we do not need to worry unless all three conditions hold: if  $A$  does not change (but  $Y$  does), then  $\Gamma(A \oplus X) = Y$  fails for all  $X$  extending  $\sigma$ ; if  $Y$  does not change then the  $\Delta$  computation remains correct; and even if both  $A$  and  $Y$  change, but a new computation with  $\sigma$  (or some extension of  $\sigma$ ) does not occur, then again  $\Gamma(A \oplus X) = Y$  fails for reals  $X$  extending  $\sigma$ . In case all conditions hold, we would like to enumerate  $\sigma$  into a Solovay test  $S$  that we build.

Instead of capturing  $U^A$ , this time, for every  $n$ , we need to capture the collection of sets  $X$  such that  $\Gamma(A \oplus X) \supseteq Y \upharpoonright n$ . Thus for every  $n$  we will have an infinite list of “boxes” on which this measurement becomes finer and finer. The first obvious obstacle is that it is not enough to ensure that for every  $n$ , the sum of errors we make “on the  $n$ th column” is finite; we need the sum of these sums to be finite. Thus we need to limit even the initial box in each column. We can do this if we pick  $Y$  to be Martin–Löf random, thanks to the following result:

**Fact 5.1.** (See Miller and Yu, [18].) If  $Y$  is a Martin–Löf random set and  $\Gamma$  is a Turing functional, then there is a constant  $C$  such that for all  $n$ , the measure of the set of sets  $X$  such that  $\Gamma(X) \supseteq Y \upharpoonright n$  is at most  $C2^{-n}$ .

In relativised form, we need  $Y$  to be  $A$ -random; but we already know that  $A$  is low for Martin–Löf randomness.

Another issue is that for every  $n$ , we have  $2^n$  many possibilities for  $\tau = Y \upharpoonright n$ . Even if for every one, the  $n$ th “agent” contributes about  $2^{-n}$  (letting the constant  $C = 1$  for simplicity), the total sum may be too much. We need a further layer of delegating authority: not only from finer to coarser boxes in the same column, but also from boxes in a certain column to ones of a previous column. The scenario is the following: for some time, without a change in  $A$ , we get  $\tau_0, \dots, \tau_k$  as possibilities for  $Y \upharpoonright n$ , and for each  $i \leq k$ , we aggregate about  $2^{-n}$  much mass of  $X$ 's such that  $\Gamma(A \oplus X) \supseteq \tau_i$ . Then a change in  $A$  occurs, which means that for all  $\tau_i$  (except for the current value of  $Y \upharpoonright n$ ), we would want to throw this mass into  $S$ . Clearly this is too much. However, we note that for three distinct  $\tau_i$ 's, at least two of  $\tau_i \upharpoonright n - 1$  must be distinct as well. This means that the same phenomenon happened for the  $(n - 1)$ st agent. If we require stringent certification, that is, to certify some  $\Gamma(A \oplus \sigma) = \tau$  we also require certification, for all initial segments  $\tau'$  of  $\tau$ , of some  $\Gamma(A \oplus \sigma') = \tau'$  for some  $\sigma' \subseteq \sigma$ , then the responsibility for all but at most one of the  $\tau_i$ 's

<sup>5</sup> Recall that a *Solovay test* is a c.e. collection  $G$  of intervals  $[\sigma]$  in Cantor space such that  $\sum_{[\sigma] \in G} 2^{-|\sigma|}$  is finite. A real  $X \in 2^\omega$  is said to *pass* the test if  $X \in [\sigma]$  for only finitely many  $[\sigma] \in G$ ; otherwise,  $X$  is *covered* by the test (or *contained* by it).

falls with previous agents and so the contribution of the  $n$ th column in this case could be kept below  $2^{-n}$ . To keep the picture tidy, we assume that  $Y$  is a left-c.e. real, so that a cancelled  $\tau_i$  will not return.

There is one last problem with this strategy: it is possible that in this situation, the computations  $\Gamma(A \oplus \sigma_i) = \tau_i$  disappear, but for the responsible  $\tau'_i \subset \tau_i$ , the corresponding computation  $\Gamma(A \oplus \sigma_i) = \tau'_i$  does not disappear because its  $A$ -use is shorter. We would then not be able to charge the  $\tau_i$ -mistake to  $\tau'_i$ 's account. However, we note that so far we did not use the third “worry condition”: that a new  $\sigma_i$  computation appears. In case it does, the corresponding  $\tau'_i$  computation must be incorrect as well (using the consistency of  $\Gamma$ ), and we could make the charge we need.

*5.1. The formal construction and proof of Theorem 1.3*

Let  $A$  be strongly jump-traceable. We are given a Martin–Löf random, left-c.e. real  $Y$  (so  $Y \equiv_T \emptyset'$ ), and a Turing functional  $\Gamma$ . We already know (Theorem 1.1) that  $A$  is low for  $K$ , and so is low for Martin–Löf randomness; in other words,  $Y$  is  $A$ -random. By Fact 5.1 relativised to  $A$ , we know that there is some constant  $c^*$  such that for all  $n$ ,

$$\mu(\{X \in 2^\omega : \Gamma(A \oplus X) \supseteq Y \upharpoonright n\}) < 2^{c^*-n}.$$

Now by replacing  $Y$  by  $Y \upharpoonright [c^*, \infty)$  (and updating  $\Gamma$  accordingly) we may assume that  $c^* = 0$ .

As in Section 4, we define a functional  $\Psi$  and get a trace  $\langle T_x \rangle$  for  $\Psi^A$ . Again we only enumerate, at stage  $s$ , axioms of the form  $\Psi^\rho(x) = \rho$  where  $\rho \subset A[s]$ . Again we get a constant  $c$  such that  $\Psi = \Psi_c$ ; the trace will be bounded by a slow-growing function  $h$  such that  $h(0) = c$ .

For every  $n, k < \omega$  we have an interval of numbers  $M_{n,k}$  which we think of as a  $(k + n + c)$ -dimensional cube, each side of which has length  $2^n 2^k$ : so the size of  $M_{n,k}$  is  $2^{(n+k)(n+k+c)}$ . The function  $h$  grows sufficiently slowly so that for all  $x \in M_{n,k}$ , we have  $|T_x| < n + k + c$ .

We let

$$R_{n,k} = \{2^{-n}m2^{-k} : m = 0, 1, \dots, 2^k\} = R_{n+k} \cap [0, 2^{-n}]$$

and let

$$R_{n,k}^+ = R_{n,k} \setminus \{0\}.$$

The coordinates of  $M_{n,k}$  are pairs  $(\tau, q)$  where  $\tau \in 2^n$  and  $q \in R_{n,k}^+$ . The idea is that for every  $\tau \in 2^n$  we have a box  $M_{\tau,k}$ ;  $M_{n,k}$  is their product. We index the elements of  $M_{n,k}$ :

$$M_{n,k}[0] = \{x_f : f : (n + k + c) \rightarrow 2^n \times R_{n,k}^+\}.$$

At stage  $s$ , we have some  $d_{n,k}[s] < (n + k + c)$  and a function  $g_{n,k}[s] : d_{n,k} \rightarrow 2^n \times R_{n,k}^+$  which gives us the current value of  $M_{n,k}$ :

$$M_{n,k}[s] = \{x_f \in M_{n,k}[0] : g_{n,k}[s] \subset f\}.$$

For  $\tau \in 2^n$ , we let

$$M_{\tau,k}[s] = \{x_f \in M_{n,k}[s] : \text{for some } q \in R_{n,k}^+, f(d_{n,k}) = (\tau, q)\}.$$

For  $\tau \in 2^n$  and  $q \in R_{n,k}^+$ , we let

$$N_{\tau,k}(q)[s] = \{x_f \in M_{n,k}[s]: f(d_{n,k}) = (\tau, q)\}.$$

For any  $\rho \in 2^{<\omega}$  and  $\tau \in 2^{<\omega}$ , we let

$$\mathcal{W}_\tau^\rho = \{X \in 2^\omega: \Gamma(\rho \oplus X) \supseteq \tau\}.$$

If  $\tau \subseteq \tau'$  and  $\rho \subseteq \rho'$  then  $\mathcal{W}_{\tau'}^{\rho'} \subseteq \mathcal{W}_\tau^\rho$ .

Like the universal machine, we assume that is a “nice” functional: any computation  $\Gamma(\rho \oplus \sigma) = \tau$  runs in at most  $\min\{|\sigma|, |\rho|\}$  many steps, and so:

- $|\tau| \leq |\rho|, |\sigma|$ ;
- If  $\sigma$  is minimal such that  $\Gamma(\rho \oplus \sigma) = \tau$  then  $|\sigma| \leq |\rho|$ .

It follows that  $\mathcal{W}_\tau^\rho$  is a clopen set which is thus presented by a finite antichain  $W_\tau^\rho$  (which means that  $\mathcal{W}_\tau^\rho = \{X : \exists \sigma \in W_\tau^\rho (\sigma \subset X)\}$ ); the map  $(\rho, \tau) \mapsto W_\tau^\rho$  is computable.

We let  $\theta_\tau^\rho = \mu(\mathcal{W}_\tau^\rho)$ . We may further assume that for all  $\rho$  and all  $\tau$ ,  $\theta_\tau^\rho < 2^{-|\tau|}$ . For if this fails for some  $\rho$  and  $\tau$ , then we know that either  $\rho \not\subset A$  or  $\tau \not\subset Y$ . In this case (assuming that  $\Gamma$  computations are given in some order), we ignore all  $\Gamma$  computations that would put some  $\theta_\tau^\rho$  beyond its permissible limit  $2^{-|\tau|}$ ; we will not lose any  $X$  such that  $\Gamma(A \oplus X) = Y$ .

The assumptions on  $\Gamma$  imply that for any  $\tau$ ,  $\theta_\tau^\rho$  is an integer multiple of  $2^{-|\rho|}$ . Also, if  $\tau \neq \langle \rangle$  then  $\mathcal{W}_\tau^\langle \rangle$  is empty and so  $\theta_\tau^\langle \rangle = 0$ .

For any  $\tau \in 2^{<\omega}$  and rational  $q$ , for any  $\nu$  such that  $\theta_\tau^\nu \geq q$ , we let  $\varrho_\tau^\nu(q)$  be the shortest  $\rho \subseteq \nu$  such that  $\theta_\tau^\rho \geq q$ .

*Construction*

At the beginning of stage  $s$  of the construction, we are given  $A[s]$  and  $T_x[s]$ .

At the beginning of the stage, the boxes  $\langle M_{n,k} \rangle$  will be in the *standard configuration* for the stage, as follows. Fix  $n \leq s$ ,  $\tau \in 2^n$ , some  $k \leq s$  and some  $q \in R_{n,k}^+$ .

- If  $q \leq \theta_\tau^{A[s] \upharpoonright s}$  then for all  $x \in N_{\tau,k}(q)[s]$  we have  $\Psi^\rho(x) \downarrow = \rho[s]$  where  $\rho = \varrho_\tau^{A[s] \upharpoonright s}(q)$ .
- If  $q > \theta_\tau^{A[s] \upharpoonright s}$  then for all  $x \in N_{\tau,k}(q)[s]$  we have  $\Psi^{A[s]}(x) \uparrow [s]$ .

Further, for all pairs  $(n, k)$  such that  $k > s$  or  $n > s$ , for all  $x \in M_{n,k}[s]$ , no definition of  $\Psi(x)$  (for any oracle) was ever made.

Let  $\rho \subseteq A[s] \upharpoonright s$ . We say that  $\rho$  is *semi-confirmed* at some point of stage  $s$  if at that point, for all  $x$  such that  $\Psi^\rho(x) \downarrow = \rho[s]$  we have  $\rho \in T_x$  at that point. We say that  $\rho$  is confirmed if every  $\rho' \subseteq \rho$  is semi-confirmed. Again, the empty string is always confirmed.

At the beginning of stage  $s$ , we speed up the enumeration of all sets to get their versions  $A[s+1], T_x[s+1]$ , so that for all  $\rho \subseteq A[s] \upharpoonright s$ , either  $\rho$  becomes confirmed or is no longer an initial segment of  $A[s+1]$ .

Next, for any  $n < s$  and  $k < s$ , we look for some  $\tau \in 2^n$  and some  $q \in R_{n,k}^+$  such that  $q \leq \theta_\tau^{A[s] \upharpoonright s}$  and such that for  $\rho = \varrho_\tau^{A[s] \upharpoonright s}(q)$  we have that  $\rho$  was confirmed at the beginning of the stage, but  $\rho \not\subset A[s+1]$ . If there are such  $\tau$  and  $q$  then we pick one such pair and extend  $g_k$  by setting  $g_k[s+1](d_k[s]) = (\tau, q)$ . Thus  $M_{n,k}[s+1] = N_{\tau,k}(q)[s]$ .

Finally, we define  $\Psi$  as necessary so that the standard configuration will hold at the beginning of stage  $s + 1$ .

The justification for why the construction runs smoothly is identical to that of the previous section.

*Verification*

Again let  $\rho^*[s]$  be the longest common initial segment of both  $A[s] \upharpoonright s$  and  $A[s + 1]$ .

We define a “functional”  $\Delta$ —it will not be quite consistent, because of us believing false  $\Gamma(A \oplus \sigma)$  computations. We let

$$\Delta = \{(\sigma, \tau) : \exists s (\tau \subset Y[s] \text{ and } \sigma \in W_\tau^{\rho^*[s]})\}.$$

For any  $X \in 2^\omega$ , we let

$$\Delta^X = \{\tau : \exists \sigma \subset X [(\sigma, \tau) \in \Delta]\}.$$

As  $\Delta$  may be inconsistent in parts,  $\Delta^X$  may fail to be an element of  $2^{\leq \omega}$ . In fact, it is a tree:

**Lemma 5.2.** *For all  $X$ ,  $\Delta^X$  is closed under taking initial segments.*

**Proof.** Suppose that  $(\sigma, \tau) \in \Delta$ , witnessed by some stage  $s$ . Suppose that  $\tau' \subset \tau$ . Then certainly  $\tau' \subset Y[s]$ . Because  $\sigma \in W_\tau^{\rho^*[s]}$ ,  $[\sigma] \subset \mathcal{W}_\tau^{\rho^*[s]}$ ; we know that  $\mathcal{W}_\tau^{\rho^*[s]} \subseteq \mathcal{W}_{\tau'}^{\rho^*[s]}$  and so there is some initial segment  $\sigma'$  of  $\sigma$  in  $W_{\tau'}^{\rho^*[s]}$ ; so  $(\sigma', \tau') \in \Delta$ .  $\square$

**Lemma 5.3.** *Let  $X \in 2^\omega$  and suppose that  $\Gamma(A \oplus X) = Y$ . Then  $Y$  is a path on  $\Delta^X$ .*

**Proof.** Let  $\tau \subset Y$ . Let  $\rho \subset A$  be some finite initial segment such that  $\Gamma(\rho \oplus X) \supseteq \tau$ . Let  $s > |\rho|$  be a late enough stage so that  $\rho \subset A[s]$  (and so  $\rho \subset A[s + 1]$ ); and  $\tau \subset Y[s]$ . Then  $\rho \subset \rho^*[s]$  and so  $X \in \mathcal{W}_\tau^{\rho^*[s]}$ ; so there is some  $\sigma \subset X$  such that  $\sigma \in W_\tau^{\rho^*[s]}$ ; so  $(\sigma, \tau) \in \Delta$ .  $\square$

Of course,  $\Delta$  is c.e. and  $\Delta^X$  is computable from  $X$ , and so if  $Y$  is an isolated path on  $\Delta^X$  then  $Y \leq_T X$  and we are done. We show that if  $\Gamma(A \oplus X) = Y$  and  $Y$  is not an isolated path on  $\Delta^X$  then  $X$  fails some Solovay test  $S$ .

For any  $k \leq s$ , let  $q_{\tau,k}[s]$  be the greatest element of  $R_{|\tau|,k}$  not greater than  $\theta_\tau^{\rho^*[s]}$ ; again, as  $\theta_\tau^{\rho^*[s]}$  is a multiple of  $2^{-s}$ , we have  $q_{\tau,s}[s] = \theta_\tau^{\rho^*[s]}$  for all  $\tau$ . Also, if  $\tau \neq \langle \rangle$  then  $q_{\tau,0}[s] = 0$  because  $R_{|\tau|,0} = \{0, 2^{-|\tau|}\}$  and  $\theta_\tau^{\rho^*[s]} < 2^{-|\tau|}$ . We let  $v_{\tau,k}[s] = \varrho_\tau^{\rho^*[s]}(q_{\tau,k}[s])$ . Again if  $k < k' \leq s$  then  $v_{\tau,k}[s] \subseteq v_{\tau,k'}[s]$ ;  $\theta_\tau^{v_{\tau,s}[s]} = \theta_\tau^{\rho^*}$  and  $v_{\tau,0}[s] = \langle \rangle$ .

We get the same calculation:

**Lemma 5.4.** *Let  $\tau \in 2^{\leq s}$ . For all  $k \in \{1, 2, \dots, s\}$ ,*

$$\theta_\tau^{v_{\tau,k}[s]} - \theta_\tau^{v_{\tau,k-1}[s]} \leq 2 \cdot 2^{-|\tau|} 2^{-k}.$$

The proof is identical to that of Lemma 4.4.

Fix  $n > 0$  and  $k, t < \omega$ . Let  $\tau = (Y[t] \upharpoonright n - 1) \hat{\ } 0$ . If  $v_{\tau,k}[t] \not\subset A[t + 2]$  then we let  $S_{n,k,t}$  be the collection of all those strings  $\sigma \in 2^t$  such that

$$[\sigma] \subset \mathcal{W}_\tau^{v_{\tau,k}[t]} \setminus \mathcal{W}_\tau^{v_{\tau,k-1}[t]}.$$

Note that because both  $\mathcal{W}_\tau^{v_{\tau,k}[t]}$  and  $\mathcal{W}_\tau^{v_{\tau,k-1}[t]}$  are the union of basic clopen sets determined by strings of length at most  $t$ , the definition of  $S_{n,k,t}$  is fine enough so that

$$S_{n,k,t} = \bigcup \{[\sigma] : \sigma \in S_{n,k,t}\} = \mathcal{W}_\tau^{v_{\tau,k}[t]} \setminus \mathcal{W}_\tau^{v_{\tau,k-1}[t]}.$$

If  $v_{\tau,k}[t] \subset A[t + 2]$  then let  $S_{n,k,t} = \emptyset$ .

Again we get (Lemma 5.4 for  $\tau = (Y[t] \upharpoonright n - 1) \hat{\ } 0$ ) that for all  $n, k$  and  $t$ , the mass of  $S_{n,k,t}$ ,

$$\sum_{\sigma \in S_{n,k,t}} 2^{-|\sigma|} = \mu(S_{n,k,t}),$$

is at most  $2 \cdot 2^{-(n+k)}$ .

The following is also familiar:

**Lemma 5.5.** *Suppose that  $S_{n,k,t} \neq \emptyset$ . Then  $M_{n,k}[t + 1] \neq M_{n,k}[t + 2]$ .*

**Proof.** Let  $\tau = (Y[t] \upharpoonright n - 1) \hat{\ } 0$  and let  $q = q_{\tau,k}[t]$ . Then  $q > 0$  because otherwise  $v_{\tau,k}[t] = \langle \rangle$  and then (as  $|\tau| > 0$ ) we would have  $\theta_\tau^{v_{\tau,k}[t]} = 0$  and so  $S_{n,k,t} = \emptyset$ . Also,  $q < 2^{-|\tau|}$  because  $\theta_\tau^{q^*} < 2^{-|\tau|}$ .

As in the proof of Lemma 4.6, we get that  $v_{\tau,k}[t] = q_{\tau,k}^{A[t] \upharpoonright t}(q)$  and it is confirmed at the beginning of stage  $t + 1$ , because it is a substring of  $\rho^*[t]$ . On the other hand, we assume that  $v_{\tau,k}[t] \not\subset A[t + 2]$ . Then  $(\tau, q)$  witness  $M_{n,k}[t + 1] \neq M_{n,k}[t + 2]$ .  $\square$

Let  $S = \bigcup_{n>0, k<\omega, t<\omega} S_{n,k,t}$ . Then the mass of  $S$

$$\sum_{\sigma \in S} 2^{-|\sigma|} \leq \sum_{n,k} 2(c + n + k)2^{-(n+k)}$$

is finite (the series grows more slowly than  $\sum_m 2(c + m)m^2 2^{-m}$ ). We end with the following.

**Lemma 5.6.** *Suppose that  $\Gamma(A \oplus X) = Y$  but that  $Y$  is a non-isolated path on  $\Delta^X$ . Then there are infinitely many initial segments of  $X$  in  $S$ .*

**Proof.** Let  $t_0 < \omega$ . We will show that there is some initial segment  $\sigma$  of  $X$  in some  $S_{n,k,t}$  for some  $t > t_0$  (in other words, that  $X \in S_{n,k,t}$ ); as  $|\sigma| = t$  we would be done.

Let  $\tau_0$  be some initial segment of  $Y$  which lies to the right of  $Y[t_0]$ . By assumption, there is some  $\tau' \in \Delta^X$  which is not an initial segment of  $Y$  but  $\tau' \supset \tau_0$ . Let  $\tau_1$  be the common initial segment of  $\tau'$  and  $Y$ ; so  $\tau_1 \supseteq \tau_0$  and so has length at least  $t_0$ .

Let  $s$  be a stage which witnesses that  $\tau' \in \Delta^X$ : so  $\tau' \subset Y[s]$  and  $X \in \mathcal{W}_{\tau'}^{\rho^*[s]}$ . Also, as  $\tau_0 \subseteq \tau' \subset Y[s]$  we must have  $s > t_0$ .

Since  $Y$  is left-c.e., it must be the case that  $\tau'$  lies to the left of  $Y$ , and so  $\tau' \supseteq \tau_1 \hat{\ } 0$  (and  $Y \supset \tau_1 \hat{\ } 1$ ). Note that for all  $t \geq s$  we have  $\tau_1 \subset Y[t]$ . Let  $\tau = \tau_1 \hat{\ } 0$ .

We know that  $X \in \mathcal{W}_{\tau'}^{\rho^*[s]} \subseteq \mathcal{W}_{\tau}^{\rho^*[s]}$ . Let  $\rho$  be the shortest initial segment of  $\rho^*[s]$  such that  $X \in \mathcal{W}_{\tau}^{\rho}$ .

As  $\rho \subseteq \rho^*$ , we know that  $\rho \subset A[s]$ ,  $A[s+1]$ ; but we cannot have  $\rho \subset A$  (as  $\Gamma(\rho \oplus X) \perp Y$ ). Thus there is a stage  $t \geq s$  such that  $\rho \subset A[t]$ ,  $A[t+1]$  but  $\rho \not\subset A[t+2]$ .

Again we know that  $\nu_{\tau,0}[t] = \langle \rangle$  and that  $\mathcal{W}_{\tau}^{\rho^*[t]} = \mathcal{W}_{\tau}^{\nu_{\tau,s}[t]}$ ; so  $X \in \mathcal{W}_{\tau}^{\nu_{\tau,t}[t]}$ ; so  $\rho \subseteq \nu_{\tau,t}[t]$  and  $\rho \supseteq \langle \rangle$ . It follows that there is a unique  $k \leq t$  such that  $\nu_{\tau,k-1}[t] \subsetneq \rho \subseteq \nu_{\tau,k}[t]$ . Thus  $X \in \mathcal{W}_{\tau}^{\nu_{\tau,k}[t]} \setminus \mathcal{W}_{\tau}^{\nu_{\tau,k-1}[t]}$ .

As  $\rho \not\subset A[t+2]$ , we have  $\nu_{\tau,k}[t] \not\subset A[t+2]$ . As we noticed before,  $Y[t] \supset \tau_1$  and so  $\tau = (Y \upharpoonright |\tau| - 1) \hat{\ } 0$ . All the conditions for setting  $S_{|\tau|,k,t} \neq \emptyset$  and  $X \in S_{|\tau|,k,t}$  are now fulfilled and we are done.  $\square$

## References

- [1] Mark Bickford, Charles F. Mills, Lowness properties of r.e. sets, typed manuscript, 1983.
- [2] Gregory J. Chaitin, Information-theoretic characterizations of recursive infinite strings, *Theoret. Comput. Sci.* 2 (1) (1976) 45–48.
- [3] Rod Downey, Noam Greenberg, Strong jump-traceability II: The general case, in preparation.
- [4] Rod Downey, Noam Greenberg, Nenad Mihailović, André Nies, Lowness for computable machines, in: *Proceedings of Computational Prospects of Infinity*, World Scientific, in press.
- [5] Rod Downey, Evan Griffiths, Schnorr randomness, *Symbolic Logic* 69 (2) (2004) 533–554.
- [6] Rod Downey, Denis Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer-Verlag, in press.
- [7] Rod Downey, Denis Hirschfeldt, André Nies, Frank Stephan, Trivial reals, in: *Proceedings of the 7th and 8th Asian Logic Conferences*, Singapore, Singapore Univ. Press, 2003, pp. 103–131.
- [8] Rod Downey, Denis Hirschfeldt, André Nies, Sebastiaan A. Terwijn, Calibrating randomness, *Bull. Symbolic Logic* 12 (3) (2006) 411–491.
- [9] Rod Downey, André Nies, Rebecca Weber, Liang Yu, Lowness and  $\Pi_2^0$  nullsets, *J. Symbolic Logic* 71 (3) (2006) 1044–1052.
- [10] Santiago Figueira, André Nies, Frank Stephan, Lowness properties and approximations of the jump, *Electr. Notes Theor. Comput. Sci.* 143 (2006) 45–57.
- [11] Denis Hirschfeldt, André Nies, Frank Stephan, Using random sets as oracles, *J. London Math. Soc.*, in press.
- [12] Bjørn Kjos-Hanssen, André Nies, Frank Stephan, Lowness for the class of Schnorr random reals, *SIAM J. Comput.* 35 (3) (2005) 647–657 (electronic).
- [13] Antonín Kučera, Sebastiaan A. Terwijn, Lowness for the class of random sets, *J. Symbolic Logic* 64 (4) (1999) 1396–1402.
- [14] Leonid A. Levin, Laws on the conservation (zero increase) of information, and questions on the foundations of probability theory, *Problemy Peredači Informacii* 10 (3) (1974) 30–35.
- [15] D.W. Loveland, A variant of the Kolmogorov concept of complexity, *Information and Control* 15 (1969) 510–526.
- [16] Joseph Miller, personal communication, April 2006.
- [17] Joseph Miller, André Nies, Randomness and computability: Open questions, *Bull. Symbolic Logic* 12 (3) (2006) 390–410.
- [18] Joseph Miller, Liang Yu, On initial segment complexity and degrees of randomness, *Trans. Amer. Math. Soc.*, in press.
- [19] Jeanleah Mohrher, A refinement of low  $n$  and high  $n$  for the r.e. degrees, *Z. Math. Logik Grundlag. Math.* 32 (1) (1986) 5–12.
- [20] André Nies, Lowness properties and randomness, *Adv. Math.* 197 (1) (2005) 274–305.
- [21] P. André Nies, Reals which compute little, in: Z. Chatzidakis, P. Koepke, W. Pohlers (Eds.), *Proceedings of Logic Colloquium 2002*, in: *Lecture Notes Log.*, vol. 27, 2002, pp. 261–275.
- [22] André Nies, Non-cupping and randomness, *Proc. Amer. Math. Soc.*, in press.
- [23] André Nies, personal communication, April 2006.
- [24] André Nies, Computability and randomness, in press.

- [25] Piergiorgio Odifreddi, *Classical Recursion Theory, Vols. I and II*, North-Holland Publishing Co., Amsterdam, 1989, 1999.
- [26] Hartley Rogers Jr., *Theory of Recursive Functions and Effective Computability*, second ed., MIT Press, Cambridge, MA, 1987.
- [27] Claus-Peter Schnorr, Zufälligkeit und Wahrscheinlichkeit. Eine algorithmische Begründung der Wahrscheinlichkeitstheorie, in: *Lecture Notes in Math.*, vol. 218, Springer-Verlag, Berlin, 1971.
- [28] Claus-Peter Schnorr, Process complexity and effective random tests, in: *Fourth Annual ACM Symposium on the Theory of Computing*, Denver, CO, 1972, *J. Comput. System Sci.* 7 (1973) 376–388.
- [29] Robert I. Soare, Recursively enumerable sets and degrees, in: *A study of computable functions and computably generated sets*, in: *Perspectives in Mathematical Logic*, Springer-Verlag, Berlin, 1987.
- [30] Robert Solovay, Draft of paper (or series of papers) on Chaitin's work, unpublished notes, May, 1975, 215 pp.
- [31] Sebastiaan Terwijn, *Computability and measure*, PhD thesis, University of Amsterdam/ILLC, 1998.
- [32] Sebastiaan Terwijn, Domenico Zambella, Computational randomness and lowness, *J. Symbolic Logic* 66 (3) (2001) 1199–1205.
- [33] Domenico Zambella, *On sequences with simple initial segments*, ILLC technical report, ML-1990-05, University of Amsterdam, 1990.