October 9, 2012

PSEUDO-JUMP INVERSION, UPPER CONE AVOIDANCE, AND STRONG JUMP-TRACEABILITY

ROD DOWNEY AND NOAM GREENBERG

Abstract. There are noncomputable c.e. sets, computable from every c.e. set relative to which $\varnothing'$ is strongly jump-traceable. This yields a natural pseudo-jump operator, increasing on all sets, which cannot be inverted back to a minimal pair or even avoiding an upper cone.

1. Introduction

While interactions between measure theory and computability theory can be traced back to the 1950's through the work of de Leeuw, Moore, Shannon and Shapiro [4] and Spector [28], most of the spectacular development of these interactions has really occurred in the last decade. Foremost in this development has been the use of methods from computability theory to understand and calibrate algorithmic randomness (see, for example, Downey and Hirschfeldt [7], Nies [24] and Downey, Hirschfeldt, Nies and Terwijn [9]). What has been less apparent is the extremely fruitful interaction the other way. That is, the use of algorithmic randomness to help us to understand computation. This paper is an example of this kind of interaction. We will use methods emanating from issues around K-triviality to answer a rather longstanding question in classical computability, explicitly articulated in Coles, Downey, Jockusch and LaForte [3], but going back to the papers of Jockusch and Shore [14, 15]: we show that there is a natural pseudo-jump operator, increasing on all sets, which cannot be inverted while avoiding an upper cone. The techniques that we introduce in order to prove this result are far from what might have been tried before the development of the randomness-related concepts of the last decade. We believe that they will have many other applications.

The origins of the methods of this paper are attempts of combinatorial characterisations of the notion of K-triviality. K-triviality has turned out to be a very important concept in algorithmic randomness. This concept originated in the work of Solovay [27], and was more recently developed starting with Downey, Hirschfeldt, Nies and Stephan [8]. Although this notion is defined in terms of initial-segment complexity (a set is K-trivial if every initial segment of it is incompressible using a prefix-free machine), the celebrated work of Nies, Hirschfeldt and others shows that K triviality coincides with notions such as lowness for K, lowness for Martin-Löf randomness, lowness for weak 2-randomness, and being a base for randomness. All of these equivalent concepts express a lack of derandomisation power of an oracle with respect to some notion of algorithmic randomness: for example, a set $A \in 2^\omega$ is low for Martin-Löf randomness if every set which is Martin-Löf random remains Martin-Löf random relative to A; in other words, A cannot detect patterns in Martin-Löf random sets. We refer the reader to [7, 9, 24, 22, 23] for details of such results.

Both authors were supported by the Marsden Fund of New Zealand.

One key question we might ask is whether there is some way to characterise classes like the K-trivials in terms of computability-theoretic considerations not involving randomness but "purely discrete" concepts like the Turing jump. In this context, Terwijn [29], and then Terwijn and Zambella [30] found a new direction in this programme. They discovered that we could use what is called *tracing* to give insight into such lowness concepts. Tracing had its origins in set theory (see [26]), but in computability the concept is defined as follows.

Definition 1.1. A *trace* for a partial function $\psi\colon \omega \to \omega$ is a sequence $T = \langle T(z)\rangle_{z<\omega}$ of finite sets such that for all $z \in \operatorname{dom}\psi$, $\psi(z) \in T(z)$.

Thus, a trace for a partial function ψ indirectly specifies the values of ψ by providing finitely many possibilities for each value; it provides a way of "guessing" the values of the function ψ. Such a trace is useful if it is easier to compute than the function ψ itself. In some sense the notion of a trace is quite old in computability theory. W. Miller and Martin [19] characterized the hyperimmune-free degrees as those Turing degrees $\mathbf{a}$ such that every (total) function $h \in \mathbf{a}$ has a computable trace (the more familiar, but equivalent, formulation, is in terms of domination). Terwijn and Zambella used a uniform version of hyperimmune-freeness, called computable traceability, to characterise lowness for Schnorr randomness, thereby giving a "combinatorial" characterisation of this lowness notion.

In this paper we are concerned not with how hard it is to compute a trace, but rather, how hard it is to enumerate it.

Definition 1.2. A trace $T = \langle T(z)\rangle$ is *computably enumerable in a Turing degree* $\mathbf{a}$ if the set of pairs $\{(x,z) : x \in T(z)\}$ is c.e. in $\mathbf{a}$.

In other words, T is c.e. in $\mathbf{a}$ if uniformly in z, an oracle in $\mathbf{a}$ can enumerate the elements of $T(z)$. It is guaranteed that each set $T(z)$ is finite, and yet if T is merely c.e. in $\mathbf{a}$, we do not expect $\mathbf{a}$ to know when the enumeration of $T(z)$ ends.

To calibrate the *strength* of a c.e. trace, we use effective bounds on the size of each component of the trace. The fewer options the trace can give for a value of the function it traces, the closer we are to knowing what that value is, and so the better the trace. The bounds are known as order functions; they calibrate rates of growth of computable functions.

Definition 1.3. An *order function* is a nondecreasing, computable and unbounded function h such that $h(0) > 0$. If h is an order function and $T = \langle T(z)\rangle$ is a trace, then we say that T is an h-*trace* (or that T is *bounded by* h) if for all z, $|T(z)| \leqslant h(z)$.

In addition to measuring the sizes of c.e. traces, order functions are used to define uniform versions of traceability notions. For example, *computable traceability* mentioned above, the uniform version of hyperimmune-freeness used by Terwijn and Zambella, is defined by requiring that traces for functions in a hyperimmune-free degree $\mathbf{a}$ are all bounded by a single order function.

Zambella (see Terwijn [29]) observed that if A is low for Martin-Löf randomness then there is an order function h such that every function computable from A has a c.e. h-trace. This was improved by Nies [22], who showed that one can replace total by partial functions. In some sense it is natural to expect a connection between uniform traceability and lowness notions such as K-triviality; if every function computable (or partial computable) from A has a c.e. h-trace, for some slow-growing order function h, then the value $\psi(n)$ of any such function can be described, in a prefix-free fashion, by $\log n + \log h(n)$ many bits.

Following this, it was a natural goal to characterise K-triviality by tracing, probably with respect to a family of order functions. This problem still remains open. However, an attempt toward a solution lead to the introduction of what seems now a fairly fundamental concept, which is not only interesting in its own right, but now has been shown to have deep connections with randomness.

Definition 1.4 (Figueira, Nies and Stephan [10])**.** Let h be an order function. An oracle $A \in 2^\omega$ is *h-jump-traceable* if every A-partial computable function has a c.e. h-trace. An oracle is *strongly jump-traceable* if it is h-jump-traceable for every order function h.

Figueira, Nies and Stephan gave a construction of a non-computable strongly jump-traceable c.e. set. Their construction bore a strong resemblance to the construction of a K-trivial c.e. set. J. Miller and Nies [18] asked if strong jump-traceability and K-triviality coincided. Cholak and the authors [2] showed that the strongly jump-traceable c.e. sets form a proper subclass of the c.e. K-trivial sets. They also showed that the class formed an ideal in the c.e. degrees. This ideal was later shown to be Π^0_4 complete by Ng [20], giving an alternative proof of the proper containment, as the K-trivial c.e. sets form a Σ^0_3 ideal. One of main contributions of the paper [2] was the introduction of new combinatorial tools for dealing with the class of c.e., strongly jump-traceable sets, collectively known as the "box-promotion" technique. We remark that recently, in [6] the authors showed how to adapt this technique to the non-c.e. case by showing that all strongly jump-traceable sets are K-trivial, when there was no a priori reason that they should even be Δ^0_2.

In view of these results it might seem that strong jump-traceability might be an interesting artifact of the studies of randomness, but as it turned out, the class of c.e., strongly jump-traceable sets has been shown to have remarkable connections with randomness. Greenberg, Hirschfeldt, and Nies [11] proved that a c.e. set is strongly jump-traceable if and only if it is computable from every superlow random set, if and only if it is computable from every superhigh random set; they related such random sets with the *benign cost functions* which by work of Greenberg and Nies [12] characterise c.e., strong jump-traceability. Greenberg and Turetsky [13] complemented work of Kučera and Nies [17] and showed that a c.e. set is strongly jump-traceable if and only if it is computable from a Demuth random set, thus solving the Demuth analogue of the random covering problem, which remains open for Martin-Löf randomness and K-triviality. Other attractive spin-offs in the arena of randomness include Nies's new work on the calculus of cost functions [25]. This material is only just beginning to work itself out and we expect a lot more to grow from these ideas.

The goal of the present paper is to use strong jump-traceability to solve a long-standing question in classical computability. This question concerns what are called pseudo-jump operators.

Definition 1.5 (Jockusch and Shore [14, 15]). A *pseudo-jump operator* is a map $J\colon 2^\omega \to 2^\omega$ of the form $J(A) = A \oplus W_e^A$.[1] A pseudo-jump operator J is *everywhere increasing* if for all $A \in 2^\omega$, $J(A) >_{\mathrm{T}} A$.

Pseudo-jump operators are a generalisation of the Turing jump operator, which maps each set $A \in 2^\omega$ to A', the halting set relative to A. Jockusch and Shore's key insight was that this generalisation of the Turing jump allowed one to give simple constructions of Turing degrees of various prescribed properties. The key concept was *pseudo-jump inversion.*

Theorem 1.6 (Jockusch and Shore [14]). *Let J be a pseudo-jump operator. Then there is some non-computable c.e. set A such that $J(A) \equiv_{\mathrm{T}} \mathbf{0}'$.*

The degree $\mathbf{a}$ of the c.e. set A given by Theorem 1.6 is an instance of *inverting* the operator J. Roughly, the idea is that J explains how to relativise to any oracle the construction of a c.e. set $J(\varnothing)$. Inverting the operator J allows us, up to Turing equivalence, to view the halting problem $\varnothing'$ as the result of the construction J, relativised to some c.e. oracle. For example, applying pseudo-jump inversion to the Turing jump operator gives a non-computable *low* set, a set whose Turing jump is as simple as possible. In turn, inverting the construction of a low set yields a *high* set, an incomplete c.e. set whose Turing jump is $\mathbf{0}''$ (as high as possible for a c.e. set). Jockusch and Shore went on to give simple constructions of c.e. degrees in every level of the low_n and high_n hierarchy using pseudo-jump inversion. These methods have seen many generalisations, and have extensions to randomness, as witnessed by Nies [24, Theorem 6.3.9], and to Π^0_1 classes by Cenzer, LaForte and Wu [1], in some sense extending the Jockusch-Soare low basis theorem.

In spite of the usefulness of pseudo-jump operators, there is distinct lack of general theory concerning them, aside from the original Jockusch-Shore papers. Coles, Downey, Jockusch and LaForte [3] studied the general theory of these operators, trying to understand what techniques pseudo-jump inversion would be compatible with. Such questions were implicit in the Jockusch-Shore papers. For example, in [3] it is shown that pseudo-jump inversion is compatible with a Friedberg strategy: for any everywhere increasing pseudo-jump operator J, it is possible to construct two Turing incomparable c.e. inversions of J. However, the question of whether pseudo-jump inversion is compatible with avoiding upper cones, and even with the construction of a minimal pair, remained open. In [3], the authors construct a pseudo-jump operator J which is increasing on c.e. sets, for which inversion cannot be combined with upper cone avoidance. That is, there is a non-computable c.e. set E which is computable from all c.e. inversions of J. However, there is no reason to believe that this operator J is increasing on all sets, rather than only on the c.e. ones. The difficulty of making J increasing globally is similar to the problem of producing a degree-invariant solution for Post's problem. Moreover, this operator J is unnatural, in that it is given by a direct priority construction. Their construction was a $\mathbf{0}'''$-priority argument.

In this paper we solve this question, by showing that there is an everywhere-increasing pseudo-jump operator for which inversion cannot be combined with upper-cone avoidance. This operator is the relativisation of the construction of a non-computable strongly jump-traceable set from [10]. Thus our example is

[1] Here W_e^A is the e^{th} c.e. set relativised to A, in some fixed list of all c.e. operators.

natural. While our construction is combinatorially complex, it does not use the $\mathbf{0}'''$-priority machinery utilised in [3], and so is *logically* simpler.

Relativising this construction, let J_{SJT} be a pseudo-jump operator, everywhere increasing, such that for all $A \in 2^\omega$, $J_{\mathrm{SJT}}(A)$ is strongly jump-traceable relative to A. Because every strongly jump-traceable set is "very low" (it is K-trivial and so superlow), every inversion A of J_{SJT} must be "very high": $\varnothing'$ is K-trivial relative to A, and so A is superhigh. Inversions of J_{SJT}, namely c.e. sets relative to which $\varnothing'$ is strongly jump-traceable, were first studied by Ng in [21], where they are called *ultrahigh*.

Nies related notions of lowness, such as superlowness, K-triviality and jump-traceability, to so-called *weak reducibilities* (see [24, Ch. 8]). These are partial relativisations of these lowness notions which are made so as to obtain transitive relations on 2^ω. The best-known weak reducibility $\leqslant_{\mathrm{LR}}$ is obtained by partially relativising lowness for randomness, equivalently lowness for K. The weak reducibility corresponding to strong jump-traceability, $\leqslant_{\mathrm{SJT}}$, is obtained by relativising the complexity of the traces, but by preserving the complexity of the bounds on the traces:

Definition 1.7. For $A, B \in 2^\omega$, we let $A \leqslant_{\mathrm{SJT}} B$ if for every (computable) order function h, every A-partial computable function ψ has an B-c.e. trace bounded by h.

Nies noted that this formulation makes the relation $\leqslant_{\mathrm{SJT}}$ transitive. To see this, suppose that $A \leqslant_{\mathrm{SJT}} B \leqslant_{\mathrm{SJT}} C$; let h be a computable order function, and let ψ be an A-partial computable function. Let T be a B-c.e. trace for ψ, bounded by $\sqrt{h}$. We write $T(n) = \{\varphi(n,1), \varphi(n,2), \dots, \varphi(n, \sqrt{h(n)})\}$, where φ is a B-partial computable function. Let S be a C-c.e. trace for φ bounded by $\sqrt{h}$. Then $R(n) = \bigcup_{i \leqslant \sqrt{h(n)}} S(n,i)$ is a C-c.e. trace for ψ bounded by h.

Akin to other reducibilities, we say that a set $B \in 2^\omega$ is SJT-*hard* if $\varnothing' \leqslant_{\mathrm{SJT}} B$. That is, if for every order function h, every partial Σ^0_2 function has an B-c.e. trace bounded by h. Certainly every ultrahigh set is SJT-hard.

The main theorem of this paper is:

Theorem 1.8. *There is a noncomputable c.e. set which is computable from every* SJT-*hard c.e. set.*

Applying the result to ultrahigh sets and so to inversions of J_{SJT}, we get:

Corollary 1.9. *There is a pseudojump operator J, increasing on all sets, which cannot be inverted while avoiding any prescribed upper cone.*

A question pursued by several researchers is whether there is a minimal pair of LR-hard c.e. degrees. By a relativisation of Nies's results, a c.e. degree $\mathbf{a}$ is LR-hard if and only if $\varnothing'$ is K-trivial relative to $\mathbf{a}$. The interest in LR-hard degrees was sparked by work of Kjos-Hanssen, J. Miller and Solomon [16], who showed that a Turing degree is LR-hard if and only if it is almost everywhere dominating, a notion suggested by Dobrinen and Simpson [5]. Relativising a result of Nies's mentioned above, we see that there is an order function h such that for any LR-hard c.e. degree $\mathbf{a}$, $\mathbf{0}'$ is h-jump-traceable relative to $\mathbf{a}$. An examination of the proof of Theorem 1.8 reveals that in fact, there is an order function g and a noncomputable c.e. set E which is computable from every c.e. set relative to which $\mathbf{0}'$ is

g-jump-traceable. If we could make g grow at least as quickly as h, we would settle the question in the negative. Currently we know that for h, we can take sufficiently large constant multiples of $\log n$. The proof of Theorem 1.8 gives an order function g whose growth rate is roughly $\log\log n$. In further work with Diamondstone and Turetsky, it is shown that for g we can take $\log_2(n)/6$. The gap is excruciatingly small, but it has not been bridged yet.

As mentioned above, mixing randomness with Turing reducibility has resulted in interesting classes of c.e. degrees. The collection of c.e. degrees which lie below all SJT-hard c.e. degrees seems to be a new ideal of c.e. degrees, about which, at this point, we know very little. Further work with Diamondstone and Turetsky, which elaborates on the technique introduced here, shows that this ideal contains superhigh c.e. degrees, but that it is likely not principal. However, further research is needed, and would no doubt require a further refinement of the box-promotion technique, which is first used to deal with highness notions in this paper.

2. Minimal pairs

The proof of Theorem 1.8 is technical. Its proof is an elaboration on the proof of a weaker result: that there is no minimal pair of c.e. SJT-hard degrees. We present this simpler proof first.

2.1. Discussion.

Let A^0 and A^1 be SJT-hard c.e. sets. We enumerate a set E, which we make noncomputable and reducible to both A^0 and A^1. The noncomputability requirements are the familiar

P^e**:** $E \neq \varphi^e$,

where $\langle \varphi^e \rangle$ is an effective enumeration of all partial computable functions. These requirements are met by the Friedberg-Muchnik strategy: a requirement P^e appoints a *follower* x, waits for the follower to be *realised*, which means $\varphi^e(x)\downarrow = 0$, and then wants to enumerate x into E.

Of course, to ensure that E is computable from A^0 and from A^1, when the requirement P^e appoints x, it needs to determine *uses* u^0 and u^1 for reducing the question "$x \in E$?" to A^0 and to A^1. We are not allowed to enumerate x into E unless *both* A^0 and A^1 change below u^0 and u^1 respectively. Moreover, if, for example $A^0\restriction u^0$ changes at some stage s, we would either need to get a change in $A^1\restriction u^1$ at the same stage (or some later stage, but bounded by some computable function), and then we would have permission to put x into E; or we would need to reset u^0 and wait for another A^0 change on the new use. That is, permissions need to be more or less simultaneous.

Fortunately, the fact that A^0 and A^1 are SJT-hard means that they do have to change often; for example, they are both high. The SJT-hardness gives us a mechanism for forcing desirable changes in these sets. SJT-hardness means that for both $i = 0, 1$, the set A^i can trace any Σ^0_2 partial function ψ by a trace $T^i = \langle T^i(z)\rangle_{z<\omega}$ which is bounded by any prescribed computable order function h. That is, if $z \in \operatorname{dom}\psi$, then $\psi(z) \in T^i(z)$, and for all z, $|T^i(z)| \leqslant h(z)$.

Let us first describe how we use SJT-hardness to manufacture changes in A^0 and A^1. To define a Σ^0_2 partial function ψ, we define a uniformly computable sequence $\langle \psi_s\rangle_{s<\omega}$ (of total functions) and let $\psi(z) = y$ if for all but finitely many s we have $\psi_s(z) = y$; if there is no such y then $z \notin \operatorname{dom}\psi$. The resulting function ψ is easily seen to have a Σ^0_2 definition. Now, for $i < 2$, let T^i be an A^i-c.e. trace for ψ, bounded by some order function h that we design. Suppose that we define, at some stage, $\psi_s(z) = s$ (as we will always do), and promise to keep $\psi_t(z) = s$ at stages $t > s$, until we see that $s \in T^i_t(z)$, where T^i_t is the stage t approximation for T^i, enumerated using the oracle A^i_t. Then such a stage t must eventuate: otherwise, $\psi(z) = s$ and so $s \in T^i(z)$, and the latter fact must be witnessed at some finite stage.

Changes in A^i are caused by repeating this process more than $h(z)$ many times: at some stage s_0 we define $\psi_{s_0}(z) = s_0$, and wait for a stage $s_1 > s_0$ at which we see $s_0 \in T^i_{s_1}(z)$. We then set $\psi_{s_1}(z) = s_1$ and repeat. For a while, A^i can enumerate all the values $s_0, s_1, \ldots$ into $T^i_t(z)$, but it cannot do this more than $h(z)$ many times. Eventually, A^i needs to "make room" in $T^i(z)$ for the latest value s_m, and in order to do so, one of the earlier values s_k must be extracted from $T^i(z)$. This is done by A^i changing below the use $u = u^i(z, s_k)$ of the enumeration of s_k into $T^i(z)$.

The very basic plan, thus, is quite simple: we pick a follower x for some requirement P^e at a stage s_0, and associate it with some input z. We set $\psi_{s_0}(z) = s_0$, and wait for $s_0 \in T^i_t(z)$. We then declare the use of the reduction of $E(x)$ to each A^i to be the use $u^i_t(z, s_0)$. If later x is realised, then we prompt A^i to change by changing $\psi(z)$.

The ideal situation is when $h(z) = 1$. We call such inputs z "1-boxes"; the "box" also refers to the trace $T^i(z)$, which is thought of as a receptacle containing it elements. If there is room for only one value in $T^i(z)$, then every change in $\psi(z)$ forces a change in A^i below the use. That is, if we are working with a 1-box, then *every* request for a change must be granted. However, we note that while we may define $h(z) = 1$ for some inputs z, there can be only finitely many such inputs (there are only finitely many 1-boxes): we need to make $\lim_n h(n) = \infty$. The other constraint which drives us to use k-boxes for $k > 1$ is that in some cases, we will want to associate different followers x with different inputs z. This is important, for example, in the case that a follower x is never realised (but in some other cases as well). If the same input z is shared by x and by infinitely many other followers y_n, then each time y_n springs to action, it will redefine $\psi(z)$. This will drive to infinity the use of reducing $E(x)$ to A^i.

So we have to deal with k-boxes, inputs z with $h(z) = k$, for $k > 1$. It is no longer the case that every request for change will be accepted. If we only consider one side, this is easily dealt with, by appointing more followers. As before, we start with a follower x_0, associated with z, set $\psi_{s_0}(z) = s_0$, get $s_0 \in T^i_t(z)$ at some later stage t, and set the uses $u^i_t(x_0)$ to be the use of enumerating s_0 into $T^i_t(z)$. Then the follower is realised, and we ask for a change by setting $\psi_{s_1}(z) = s_1$. It is possible that change will not be given, but rather, both s_1 and s_0 will appear in $T^i_t(z)$. We then appoint a new follower x_1, and tie its use to the use of enumerating s_1 into T^i_t. Repeat; eventually, after no more than $k + 1$ many tries, one of the followers will be permitted.

We recall, though, that we require *simultaneous* permission from A^0 and A^1. Again, if we have 1-boxes, this is not a problem: both A^0 and A^1 have to permit every follower, so we can wait until they both permit our follower and enumerate it into E. But for $k > 1$ we have a problem. Take, for example, $k = 2$, and consider the following chain of events. We start with a follower x_0, tied to a value $s_0 \in T^i(z)$. It is realised, and we request permission from both sides. However, the sides do not act in tandem: A^1, say, gives permission (and takes s_0 out of $T^1(z)$), but A^0

does not. At that point we have $T^0(z) = \{s_0, s_1\}$ and $T^1(z) = \{s_1\}$. We cannot enumerate x_0 into E, and we need to define a new use for reducing $E(x_0)$ to A^1; the only candidate is the use of enumerating s_1 into $T^1(z)$. Note that we should not "keep the A^1 permission open", that is, refrain from attaching a new use to reducing $E(x_0)$ to A^1; it is possible that A^0 will never give permission to x_0, and so we will not be able to enumerate x_0 into E. In that case, as we shortly see, later followers will be permitted by both A^0 and A^1, so the diagonalisation will succeed, but the reduction of E to A^1 will be ruined.

However, what actually happens is the following. A new follower x_1 is tied to s_1 on both sides. When x_1 is realised, we ask for permission again (define $\psi(z) = s_2$). This time, A^0 permits *both* x_0 and x_1 but A^1 permits neither. We get $T^0(z) = \{s_2\}$ and $T^1(z) = \{s_1, s_2\}$. The next time, A^1 will permit but A^0 will not; this can repeat for ever. Essentially, our opponent is playing Lachlan's "freeze one side which changing the other" strategy for constructing a minimal pair of c.e. degrees. And indeed, up to some level, this strategy is successful; if the growth rate of h is about 2^n, then there is a minimal pair of c.e. sets which are both h-JT hard (equivalently, superhigh with truth-table reduction with exponential use); see for example Ng's [21]. We need to make use of the fact that the order function h grows slowly.

We remark that a similar phenomenon occurs close to $\mathbf{0}$, in the proof [2] that the SJT degrees form an ideal. The closure under join happens only below some slow growth rate. For every order function g there is an order function $\tilde{g}$, so that whenever two c.e. sets B_0 and B_1 are $\tilde{g}$-jump-traceable, their join $B_0 \oplus B_1$ is g-jump-traceable. However, no matter how quickly g grows, $\tilde{g}$ has to be sub-logarithmic for the argument to work, and indeed, for sufficiently quickly growing orders g (around exponential), there are two g-jump-traceable sets which join to $\varnothing'$. The key in that argument, as it will be here, is the interaction between the various levels of boxes. In both proofs, we design a *cursus honorum*, a path for followers to progress from k-boxes to $(k-1)$-boxes and so on, and if they do not get "stuck", they eventually reach 1-boxes and can use them to obtain simultaneous permission, as described above.

The key concept is that of *box promotion*, and it was already implicitly used above when we discussed how to obtain permission from one set if we do not use 1-boxes. Consider the counter-factual above for the case $k = 2$. The first follower x_0 is not permitted by A^0. This lack of permission promotes the box $T^0(z)$ to be a 1-box in the eyes of all other followers. While A^0 persists with not permitting x_0, we can win by first using at most two new followers to get another permission from A^1 for a follower $y > x$; and then get immediate permission for y from A^0. The new ingredient, compared to other box-promotion arguments, is that promotions can be undone. In our example, when A^0 does permit x_0, the box $T^1(z)$ returns to being a 2-box.

The solution is to make use of actual 1-boxes as well as 2-boxes. Going on with our example, when A^1 gives permission to x_0 but A^0 does not, we get the new use for reducing $E(x_0)$ to A^1 not from the 2-box $T^1(z)$ but from a 1-box $T^1(z')$ (where $z' < z$). We then wait for A^0-permission for x_0. If this is *ever* given, then we will immediately get A^1 permission for x_0 as well. If it is *never* given, then the box $T^0(z)$ has been *permanently* promoted to being a 1-box, and the same strategy can take place for other followers, this time using an A^1-2-box and the manufactured

A^0-1-box. And this strategy can be propagated up the levels. Even before A^1 gives the first permission to x_0, followers for weaker requirements can use z as a 1-box, this time on the A^1-side. The general process is illustrated in Figures 1 and 2.

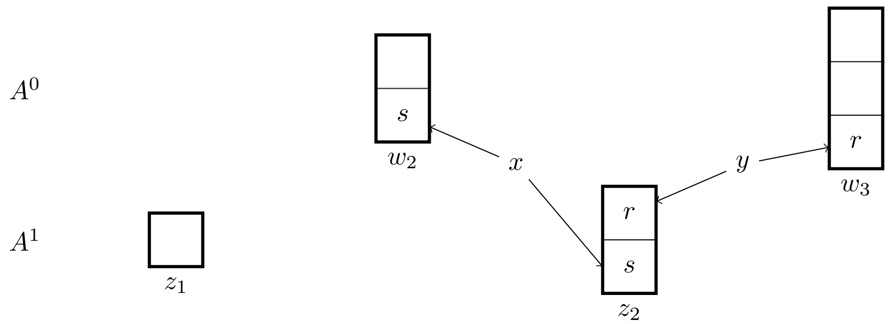

FIGURE 1. The follower x is waiting for an A^1-permission. Meanwhile, a weaker follower y can treat $\{z_2\}$ as a 1-box.

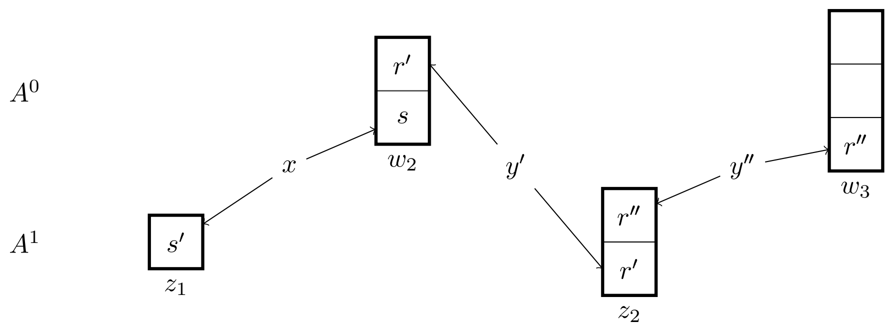

FIGURE 2. Follower x received permission from A^1 at stage s'. Its A^1-pointer moved to z_1; it is now waiting for A^0-permission. The follower y was cancelled at stage s', but while x is waiting for permission, a new follower y' is treating $\{w_2\}$ as a 1-box, and a follower y'' is treating $\{z_2\}$ as a 1-box.

This is the basic idea, but on its own it is not sufficient. We need to introduce a mainstay of the box-promotion method, namely the amplification of promotions. In a nutshell, we want a single promotion to be useful for many other followers, not only one; this is why we use aggregations of inputs, which are sometimes called *metaboxes*. Terminology is loose, though, and sometimes by "box" we also mean a metabox. The principle is that instead of associating the use for reducing $E(x)$ to A^i for some follower x with only one input z, we use a finite collection $M = M(x)$ of inputs. The use will be the maximum of the uses of enumerating the associated value s into $T^i(z)$ as z ranges over all the elements of M. Thus, an extraction of s from even one of these $T^i(z)$ will give permission to x, and so, while permission is not given, all of the boxes $T^i(z)$ for $z \in M$ are promoted by one level. A number of followers can now apportion these boxes between them.

To see why this is necessary, we need a clearer picture of the construction. The description above may give the wrong impression that when followers are appointed, they always point on one side at a (probably manufactured) 1-box. The reason this is not so is that some followers are never realised. While they are not realised, of course, no permission is useful, as we want to keep them out of E. While we wait for a follower to be realised, we need to protect its uses from being driven to infinity by the action of other followers. So unrealised followers cannot share boxes with other followers weaker than themselves. Only once they are realised they can collaborate with other followers to generate permissions.

We certainly cannot let a new unrealised follower point at a manufactured 1-box, as this could be wasteful: an unrealised follower does not help promote boxes for weaker followers. So unrealised followers for requirement P^5, say, will start pointing at 5-boxes on both sides. When they are realised and receive permission, they start "walking" down the levels, possibly getting stuck in some places, thereby contributing to the promotion of some boxes; or eventually making it to the bottom, and so making use of 1-boxes to obtain simultaneous permission. At each step we wait for permission from one side, and ignore permissions from the other. In particular, when setting a new follower up, the side it first requests permission from needs to set up a "private box" devoted only to followers for the same requirement. When the follower is realised and receives permission, it joins the rest of the construction and now cooperates with followers for other requirements in the promotion mechanism. While we digress, we note that we only use private boxes on one side. Say a new follower x first requests permission from A^1 (it will be convenient, but not important, to make it so for all followers). When it receives permission, x moves its pointer from the A^1-private box to the next level down, where it can use boxes that are involved in the general construction. However, x did not receive A^0-permission, and so cannot move its A^0-pointer; so on the A^0-side we have to start pointing at "public" boxes from the beginning.

Returning to metaboxes, now the point is that two followers, for different requirements, may be pointing at the same levels on both sides. Consider for example the situation in Figure 1 above. A follower x points to 2-boxes on either side, namely w_2 and z_2. While it is waiting for permission from A^1, other followers such as y treat z_2 as a 1-box, but on the A^0-side they point to the 3-box w_3. They are waiting for permission from A^0, on that 3-box, before they can make use of the fact that on the A^1-side they have a 1-box which gives immediate permissions. Because w_3 is a 3-box, we need at least two followers y_0 and y_1 pointing at it in order to make it a 1-box and keep the process going upwards. For this, it is of course essential, that y_0 and y_1 are both pointing at the same input w_3 on the A^1-side: they have to share the box, as this is the only way they can combine forces to make a useful promotion. However, these two followers do not want to share the 1-box on the A^1-side, because again, this may result in the use of say y_0 being driven to infinity. Thus, we make the 2-box on the A^1-side a metabox; see Figure 3.

Of course, we need to apply this reasoning to every level: thinking toward the 4-boxes, we may need to split the collection of 3-boxes into at least three disjoint parts. But it is not sufficient to have only k many k-boxes.[2] Consider again the

[2] We remark at this point that it would be *extremely nice* to have a version of this construction requiring only k many k-boxes. As indicated in the introduction, this would imply that there is no minimal pair of LR-hard c.e. sets.

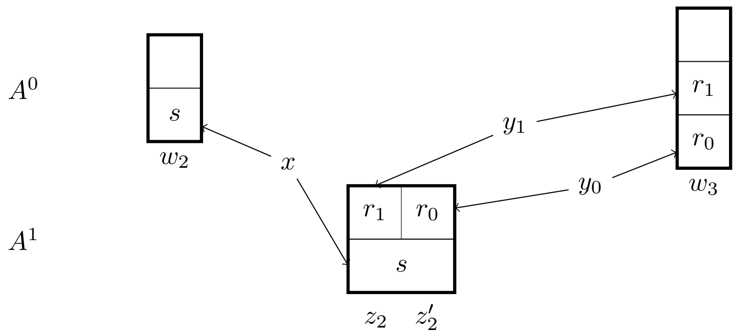

FIGURE 3. The follower x is pointing, on the A^1-side, at the metabox $\{z_2, z_2'\}$; y_0 is pointing at z_2' and y_1 at z_2, while on the A^0-side, they share w_3.

case of 2-boxes. Even before any promotions are made to 2-boxes, we may need disjoint 2-boxes (on the A^1 side) for two followers y_0 and y_1 which are waiting for permission from A^0 on 3-boxes. Either one of these may need to be split up in the future: at most one, but we cannot tell in advance which one. Suppose that y_0 is stronger than y_1. If y_0 gets permission from A^0, and its A^0 pointer moves from w_3 to w_2, then y_1 is cancelled, the part of $M(2)$ which was pointed to by y_0 gets promoted, and it will be split up between future followers pointing at $M(2)$ and w_3. The part of $M(2)$ which was pointed to by y_1 is not promoted, and will not be used until y_0 is cancelled or moved. If y_1, but not y_0, gets permission, then y_0 is not cancelled, but the part of $M(2)$ to which it is pointing is not promoted, and will not be used by other followers; a future follower will point to sub-boxes of the metabox pointed to by y_1. See Figures 4, 5, and 6. So we need at least three 2-boxes, and in general about k^k-many k-boxes. This explains why the order function h needs to grow slowly.

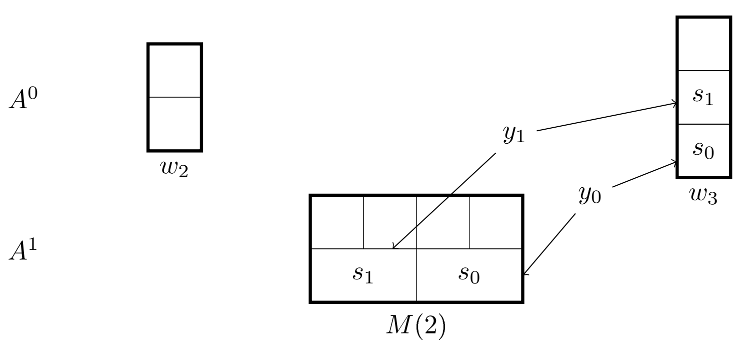

FIGURE 4. Both y_0 and y_1 seek permission from A^0. We do not know which, if either, will be permitted.

We covered the main ideas which make the construction work, and in the next subsection we give the full details. But before we do so, we remark on one other

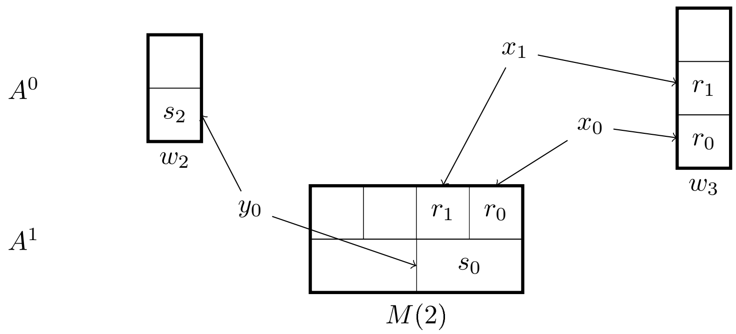

FIGURE 5. The follower y_0 was permitted. Followers x_0 and x_1 use y_0's boxes. y_1 is cancelled.

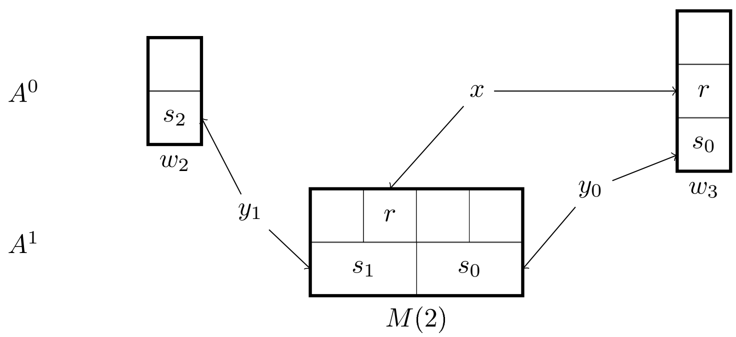

FIGURE 6. In an alternate reality, y_1 was permitted. A new follower x uses y_1's boxes.

complication. We explained earlier why we can start with only finitely many 1-boxes; h needs to be unbounded and nondecreasing. However, it is possible that we will have access to no 1-boxes at all. The reason is "overhead" (or "tax") charged by the recursion theorem (and when proving the full theorem in the next section, the reason would be the need to guess traces for the functions that we build). At the beginning we said that we define a function ψ and work with traces T^i for ψ. Circularity is avoided by using the recursion theorem, but formally, for the recursion theorem to work, we need to describe infinitely many constructions, and the construction indexed by the fixed point given by the recursion theorem is the one which will work. All of these versions of the construction need to cooperate in defining the order function h and cannot interfere with each other. To make $\lim h(n) = \infty$, the correct construction, indexed by a constant c, will only have access to k boxes for $k \geqslant c$.

Now it would seem that the very basis of the construction that we described has disappeared. Our followers progress down the levels, and if they do not get stuck, then they should eventually reach 1-boxes and receive simultaneous permission. We then argue that promotions imply that only finitely many followers for each requirement get stuck along the way. But if there are no 1-boxes, there seems to be no source for simultaneous permissions.

And indeed we cannot quite get simultaneous permissions. When a follower makes it all the way down, is pointing at c-boxes on both sides, and receives permission from one side, say A^1 (it does not matter which, as long as we choose one), then the only thing that we can do is leave this permission open. The argument showing that only finitely many followers get stuck at each level, will show that there are only finitely many followers for which we leave an open A^1-permission and do not later get an A^0-permission as well. Thus, our reduction procedure to A^1 will converge for all but finitely many numbers. This is the only source for non-uniformity in our construction: the index for E and the index for a reduction of E to A_0 are computed effectively from indices for A^i and for the universal A^i-traces. This "near uniformity" of the construction turns out to be important when we prove Theorem 1.8, when we try to generalise the construction we discussed to deal with infinitely many oracles. Looking ahead, as an exercise, the reader is invited to try to show that if A^0, A^1 and A^2 are SJT-hard c.e. sets, then there is a noncomputable c.e. set E reducible to each of A^0, A^1 and A^2.

2.2. **Construction.** We can now give the details of the construction. For tidiness, we define two partial Σ^0_2 functions, ψ^0 and ψ^1, by defining uniformly computable sequences $\langle \psi^i_s \rangle$ of total functions, and let ψ^i be the partial limit of the sequence $\langle \psi^i_s \rangle$. For both $i < 2$, we let $\psi_0(n) = 0$ for all n. At stage s we may define $\psi^i_s(z) = s$ for some z's; for all other z's, we let $\psi^i_s(z) = \psi^i_{s-1}(z)$. We let $\psi^i(z) = \lim_s \psi^i_s(z)$ if the limit exists; otherwise $\psi^i(z)\uparrow$.

We shortly define an order function h. We claim that we can obtain a constant $c \geqslant 1$, and for both $i < 2$, A^i-c.e. traces T^i for ψ^i, which are bounded by h. The definition of h will rely on the constant c. That is, for each $c < \omega$ we define an order function h_c, with the restriction that $h_c \geqslant c$. From these we obtain an order function h_* with the property that for all c, x and $i < 2$, $h_*(c, i, x) \leqslant h_c(x)$; here $h_c \geqslant c$ is used. For $i < 2$ we take A^i-c.e. traces T^i_*, bounded by h_*, for the Σ^0_2 function $(c, i, x) \mapsto \Phi_c(\varnothing', i, x)$. From this we obtain, uniformly in c, A^i-c.e. traces T^i_c for $x \mapsto \Phi_c(\varnothing'; i, x)$ bounded by h_c. The recursion theorem gives us an index c such that each ψ^i is the function $x \mapsto \Phi_c(\varnothing'; i, x)$. We then take $h = h_c$ and $T^i = T^i_c$.

So we fix c, and define h (and recall that we must have $h \geqslant c$). For $k \geqslant c$, we define $I(k)$, which are finite intervals of natural numbers. These are successive intervals, so to define these intervals, it suffices to determine their size, which we set to $|I(k)| = 1 + (2k + 2)^{k+1}$. This, in turn, determines h, because we define $h(z) = k$ for all $z \in I(k)$.

Let $i < 2$. We let $\langle A^i_s \rangle$ be an effective enumeration of A^i. For $z < s$, we let $T^i_s(z)$ be the collection of numbers enumerated into $T^i(z)$ with oracle A^i_s. By delaying enumeration of elements into $T^i_s(z)$, we may assume that for all s, $|T^i_s(z)| \leqslant h(z)$.

For $t \in T^i_s(z)$, we let $u^i_s(z, t)$ be the A^i_s-use of enumerating t into $T^i_s(z)$. Similarly, for $t \in T^i(z)$ we let $u^i(z, t)$ be the A^i-use of enumerating t into $T^i(z)$.

Now by the fact that T^i traces ψ^i, we may assume that for all s, there is some $t > s$ such that for all $z < s$, for both $i < 2$, $\psi^i_s(z) \in T^i_t(z)$. To ensure this, when we redefine $\psi^i_s(z)$, we search for a stage $t > s$ as is sought after. While we search for t, we hold the definition of ψ^i, so that if such a stage is not found, we have $\psi^i = \psi^i_s$. But then we have a contradiction to the fact that T^i traces ψ^i. [We note that this complies with the requirement of the recursion theorem, that for all values of c, we give a Σ_2^0 index for the functions ψ^i, not just for the "true" version of the construction.]

There are two options for handling this fact. We could speed up the enumeration of the sets A^0 and A^1 and assume that the stage t provided by the claim equals s. This necessitates a cascading effect: a typical response for finding $s \in T^i_s(z)$, which likely involves an A^i-change, is to redefine $\psi^i_s(z') = s$ for more values z', and so a further speed-up of A^0 and A^1 so that $s \in T^i_s(z')$ for these inputs z', and so on. We can argue that at each stage, this process repeats only finitely many times.

We do not take this approach. Mostly, this is because we want to present a construction which is close to the full construction proving Theorem 1.8. In that construction, we work with all c.e. oracles, not only with two c.e. oracles A^0 and A^1 which are guaranteed to be SJT-hard. So in the full construction, we need to guess which traces, if any, trace the functions ψ^i we build; for some oracles no guess will be correct. In other words, in the full construction, the fact above is only guaranteed to hold for oracles which are SJT-hard, and only for correct guesses of their trace. Hence in the full construction, we cannot speed up all sets and get instant gratification. We have to restrict ourselves to stages at which our guesses seem correct.

We apply a similar, more patient approach in the current construction. Following Nies's terminology, the construction will take place at a computable set of *stages*. Given a *stage* s, we let the following *stage* be the next stage $t > s$ such that for all z which were encountered by stage s, $\psi^i_s(z) \in T^i_t(z)$. Between stages s and t, no change is made to ψ^i, or to any other object of the construction. The fact above now says that there are infinitely many *stages*.

Followers. We try to meet the requirements P^e for $e \geqslant c$. As mentioned above, a requirement P^e appoints *followers*. A follower x for P^e is *realised* at stage s if $\varphi^e_s(x) = 0$. The requirement P^e is *satisfied* at stage s if there is some $x \in E_s$ which is realised for P^e. If P^e is satisfied at stage s, then P^e takes no action at stage s.

We say that a follower x is *alive* at the end of a stage s if x was appointed at some stage $s' \leqslant s$ and was not cancelled at any stage $s'' \in (s', s]$ (no follower is appointed more than once). Recall that in order to reduce the question "$x \in E$?" to both A^0 and A^1, we tie the uses of these reduction to A^i-uses of enumerating elements into trace components $T^i(z)$ for a collection of inputs z. If a follower x is alive at the end of stage s, then we will define for $i = 0$ and usually also for $i = 1$:

- $M^i_s(x)$: this is the finite collection of inputs z to which x is tied at the end of stage s; and
- a value $t^i_s(x)$: this is the value that we use to determine the use.

These will be defined for both $i = 0$ and $i = 1$, except for the case that we decide to leave an A^1-permission for x open; then, only $M^0_s(x)$ and $t^0_s(x)$ are defined. If $M^i_s(x)$ and $t^i_s(x)$ are defined, then we say that x "points" at $M^i_s(x)$.

For $i < 2$, if $M^i_s(x)$ and $t^i_s(x)$ are defined, and further, for all $z \in M^i_s(x)$ we have $t^i_s(x) \in T^i_s(z)$, then we let

$$v^i_s(x) = \max\left\{u^i_s\left(z, t^i_s(x)\right) \,:\, z \in M^i_s(x)\right\},$$

where recall that $u^i_s(z,t)$ is the A^i_s-use of enumerating t into $T^i_s(z)$. The value $v^i_s(x)$ is the use for reducing $E(x)$ to A^i at stage s. In the case that $v^i_s(x)$ is defined, we say that *an i-computation is defined for x at stage s*, and write $v^i_s(x)\downarrow$. If no such computation is defined, then x has permission from A^i to enter E; we write $v^i_s(x)\uparrow$.

The main technical part of the construction will be the choice of $M^i_s(x)$ (and $t^i_s(x)$) for followers x. Recall that the idea is to for x to "zig-zag" its way down to more and more powerful boxes. Thus, in order to define $M^i_s(x)$ we will first define a *level* at which x points at stage s, and denote that level by $k^i_s(x)$. This will be a number between c and e (where x is a follower for P^e). The meaning of x pointing at level $k^i_s(x)$ is that we will define $M^i_s(x) \subset I(k^i_s(x))$. The point is that for all $z \in M^i_s(x)$, $|T^i_s(z)| \leqslant k^i_s(x)$. The smaller $k^i_s(x)$, the easier will it be to generate A^i permissions for x; but there will be fewer available boxes, and so we will need to take care which followers get access to "stronger" boxes.

Further, if both $M^0_s(x)$ and $M^1_s(x)$ are defined, then we will prioritise one of these. We will define another parameter $\mathtt{top}_s(x) \in \{0,1\}$. Let $i = \mathtt{top}_s(x)$; we will next seek permission from A^i, and ignore the other set, using permissions from A^{1-i} only if they are accompanied by a permission from A^i (in which case, of course, we can enumerate x into E). If only permission from A^i is given, then we will use this permission to let x progress, so we decrease the level $k^i_s(x)$ by one (and redefine $M^i_s(x)$); and change the "top" to be $1-i$. If $M^1_s(x)$ is not defined (A^1-permission is left open), then $\mathtt{top}_s(x) = 0$, of course. For i, s and x, either all of the parameters $M^i_s(x), t^i_s(x), k^i_s(x)$ and $\mathtt{top}_s(x)$ are defined, or none of them are.

We can now properly define when a follower is permitted. Let $s > 0$ be a *stage* at the *beginning* of which a follower x is alive. Let r be the last *stage* before s; so x was alive by the end of stage r. Let $i = \mathtt{top}_r(x)$. We say that x is *permitted* at stage s if $v^i_r(x)\uparrow$, or if $A^i_s \restriction v^i_r(x) \neq A^i_r \restriction v^i_r(x)$.

All followers alive at stage s are linearly ordered by priority, which is determined by the stage of their appointment (at most one follower is appointed at each *stage*). As usual, when a new follower is appointed, it is chosen to be large relative to any number previously encountered in the construction, and so if x and y are followers at a stage s, then x is stronger than y if and only if $x < y$.

We say that a follower x *requires attention* at *stage* s if it is appointed at stage s, or it is permitted at stage s. If some follower x requires attention at stage s, then the strongest such follower will *receive attention.* If a follower x, alive both at the beginning and at the end of a stage s, does not receive attention at stage s, then there is no change to x's parameters $M^i(x), t^i(x), k^i(x)$ and $\mathtt{top}(x)$ from the previous *stage* .

If a follower x receives attention at stage s, then all followers weaker than x are cancelled. In addition, when a requirement P^e appoints a new follower, all followers for all weaker requirements are cancelled. As a result, if x is a follower for a requirement P^e, and y is a follower for a weaker requirement $P^{e'}$, both alive at some stage, then x is stronger than y.

Carving the boxes. Our next task is to explain exactly how each $M^i_s(x)$ is chosen.

For $k \geqslant c$, let $a(k) = 2k+2$; so $|I(k)| = 1 + a(k)^{k+1}$. We first let $J(k) = \{\min I(k)\}$; we will have $M^i_s(x) = J(e)$ for a new follower for P^e (this will be the "private box" for x, where it is interrupted by no other followers.)

Then, we break up $I(k) - J(k)$ into a tree of sub-boxes. These will be indexed by strings $\beta \in {}^{\leqslant k+1}a(k)$, that is, strings β of length at most $k+1$, which mention only numbers smaller than $a(k)$. For each such string we define a sub-box $B(k,\beta) \subseteq I(k) - J(k)$. They form a tree in the sense that $B(k,\alpha) \subseteq B(k,\beta)$ if α extends β, and $B(k,\alpha) \cap B(k,\beta) = \varnothing$ if α and β are incomparable strings. Other than the private boxes, each $M^i_s(x)$ will be one of the boxes $B(k,\beta)$ (for $k = k^i_s(x)$, the level at which x points at stage s). And it is important to note that each $B(k,\beta)$ is disjoint from $J(k)$.

The sub-boxes $B(k,\beta)$ are defined by induction on the length of β. We start of course with $B(k,\langle\rangle) = I(k) - J(k)$, where $\langle\rangle$ is the empty string. Note that $|B(k,\langle\rangle)| = a(k)^{k+1} = a(k)^{k+1-|\langle\rangle|}$; in general, we will have $|B(k,\beta)| = a(k)^{k+1-|\beta|}$. Other than that, the actual definition of $B(k,\beta)$ does not matter; given $B(k,\beta)$, with $|\beta| \leqslant k$, we let $\{B(k,\beta\hat{\ }m) : m < a(k)\}$ be a partition of $B(k,\beta)$ into $a(k)$ many subsets of equal size.

Let x be a follower, alive at the end of some stage s. Let $i < 2$ and suppose that x's parameters are defined. Let $k = k^i_s(x)$. As mentioned above, either:

- $M^i_s(x) = J(k)$ (we say that *x points to its private box*); or
- $M^i_s(x) = B(k,\beta)$ for some string β.

In the latter case, we let $\alpha^i_s(x)$ denote that string β. So in this case, specifying the box $M^i_s(x)$ is the same as specifying the string $\alpha^i_s(x)$.

We decree that every follower starts by first asking for permission from A^1 (this is not important, the followers just need to start somewhere) and so the private boxes will actually only be used on the A^1-side. That is, if x is appointed at stage s then we will have $\mathtt{top}_s(x) = 1$ and $M^1_s(x) = J(k)$ (while $M^0_s(x)$ will be "public", i.e. some $B(e,\beta)$). If x later moves (which will happen only after it is realised), then the first move will be away from the private box $J(k)$, so after that, we will have $\alpha^i_t(x)$ defined on both sides.

Assigning the boxes. We now need to explain how we choose the strings $\alpha^i_s(x)$. Suppose that a follower x receives attention at stage s and we allow it to move. It just received permission from A^i for $i = \mathtt{top}_r(x)$ (where again $r < s$ is the previous *stage*). We will set $\mathtt{top}_s(x) = 1 - i$. The box $M^{1-i}_r(x)$ is already defined, and will not be changed ($M^{1-i}_s(x) = M^{1-i}_r(x)$). However, we want to use the permission and move the i-pointer for x from level $k^i_r(x)$ to one level down, so we will let $k^i_s(x) = k^i_r(x) - 1$; for brevity, denote this level now by k. We need to define $M^i_s(x)$, which amounts to choosing $\alpha^i_s(x)$ to be some string $\beta \in {}^{\leqslant k+1}a(k)$.

This choice arises from considering the interaction of x with other followers y which are still alive at the end of stage s (so are stronger than x) which also point to level k on the A^i-side (i.e. $k^i_s(y) = k$). There are two kinds of such followers y: ones for which also $\mathtt{top}_s(y) = i$, and ones for which $\mathtt{top}_s(y) = 1-i$. Since all the followers follow the same path, in the first case we will have $k^{1-i}_s(y) = k^{1-i}_s(y) - 1$ (y is already one step ahead of x), and in the second case we will have $k^{1-i}_s(y) = k^{1-i}_s(x)$, so x and y point at the same levels on both sides.

Recall that the combinatorics of the construction (and so our choice of $\alpha^i_s(x)$) are driven by the competition, between followers, for resources – boxes. We ultimately will need to show that there are *any* boxes

There are two principles by which we have to abide:

(1) If $\mathtt{top}_s(y) = i$ then $M^i_s(x) \subset M^i_s(y)$.
(2) If $\mathtt{top}_s(y) = 1 - i$, then $M^i_s(x)$ and $M^i_s(y)$ must be disjoint.

To explain why, we need to observe the intended combinatorics of the construction. One main question is: how do we arrange that sufficiently many followers eventually do receive permission, so that all requirements will be met? Recall that this involves "pressing the button", i.e. changing $\psi^i(z)$, often enough, so that the traces $T^i_s(z)$

get filled up and eventually force A^i to give permissions. We execute this plan by redefining, at the stage s at which x moves, $\psi^i_s(z) = s$ for all $z \in M^i_s(x)$; we then let $t^i_s(x) = s$. If we obey the first principle above, then the intersection of $M^i_s(y)$ for *all* followers y for which $k^i_s(y) = k$ and $\mathtt{top}_s(y) = i$ will be nonempty, and the values $t^i_s(y)$ for these followers y will be distinct. This will imply there can be no more than k many such followers y. In turn, this will give us a bound on the number of boxes required overall at the next level down on the other side A^{1-i}, and will show that we can choose appropriate strings $\alpha^{1-i}_t(y)$ for such followers y.

The second principle follows from the first. Consider a follower y with $k^i_s(y) = k$ and $\mathtt{top}_s(y) = 1-i$. It is possible that y will never receive permission again. Consider what could happen if we let $M^i_s(x)$ and $M^i_s(y)$ intersect. Let z be in the intersection. As mentioned, at stage s we let $\psi^i_s(z) = s$. This prompts an enumeration into $T^i(z)$ and a possible change in A^i, which will give an A^i-permission to y. But this permission is useless; y is waiting for permission from A^{1-i}, not from A^i. So y will not move; but it must update its use $v^i_s(y)$ to the new use of enumerating s into $T^i(z)$ – otherwise, in the case that y eventually will move and require A^i-permission, it will not be in position to get one. Later, x moves away or is cancelled, and a new follower x' arrives and acts as x did. This can happen infinitely often, and will drive the use $v^i(y)$ to infinity, ruining the reduction of $E(y)$ to A^i.

The two principles dictate how to choose $M^i_s(x)$. We find a weakest y with $k^i_s(y) = k$ and $\mathtt{top}_s(y) = i$, and require $M^i_s(x) \subset M^i_s(y)$, so $\alpha^i_s(x)$ to extend $\alpha^i_s(y)$; we let it be a 1-bit extension: $\alpha^i_s(x) = \alpha^i_s(y)\hat{\ }m$ for some $m < a(k)$. So we need to make sure that $|\alpha^i_s(y)| \leqslant k$. Of course, if there is no such y then we replace $\alpha^i_s(y)$ by the empty string $\langle\rangle$. Then, we need to find a value $m < a(k)$ such that no $\alpha^i_s(w)$, for followers w with $k^i_s(w) = k$ and $\mathtt{top}_s(w) = 1-i$, is comparable with $\alpha^i_s(y)\hat{\ }m$; it will be sufficient to show that fewer than $a(k)$ many such followers w exist, since the worst case is that they all use $\alpha^i_s(y)\hat{\ }m'$ for some m'. Proving this bound on the number of such followers w will be most of our work, and as already mentioned, it will essentially rely on the linear structure of the boxes for such w on the A^{1-i}-side, and the bound on the size of the traces at that side.

We formalise the discussion above by some more notation. Let s be a *stage* , and suppose that at stage s, a follower x receives attention (is either appointed or moves); let $i < 2$ and $k \geqslant c$ be such that we want to define $k^i_s(x) = k$ and define $M^i_s(x)$. At stage s we cancel all followers weaker than x, so the only followers alive at the end of the stage are x and followers stronger than x.

We define a string $\beta^i_s(k)$, the intended parent of the new $\alpha^i_s(x)$:

- Suppose that there is some follower $y \neq x$, alive at the end of stage s, such that $k^i_s(y) = k$, $\mathtt{top}_s(y) = i$, and $M^i_s(y) \neq J(k)$. Take y to be the weakest such follower, and let $\beta^i_s(k) = \alpha^i_s(y)$.
- If there is no such follower y, let $\beta^i_s(k) = \langle\rangle$.

Further, given $\beta^i_s(k)$, let $m^i_s(k)$ be the least $m < a(k)$ such that there is no follower $w \neq x$, alive at the end of stage s, such that $k^i_s(w) = k$ and $\alpha^i_s(w) = \beta^i_s(k)\hat{\ }m$.

The following statement is the formalisation of the fact that a string $\alpha^i_s(x)$ can be chosen as required at stage s:

$\otimes_s$**:** $|\beta^i_s(k)| \leqslant k$, and a number $m^i_s(k)$ as above exists.

We will carry $\otimes$ along with the construction. That is, at stage s of the construction we will assume, inductively, that $\otimes_s$ holds; we then carry out the instructions based on this assumption, and then show that $\otimes_{s+1}$ holds as well.

Before we give the formal construction, we remind the reader of the notation introduced so far.

- ψ^i: the Σ^0_2 partial function used to irritate the c.e. set A^i. T^i: an A^i-c.e. trace for ψ^i.
- Parameters of follower x for some requirement, alive at a stage s; $M^i_s(x)$: the box to which x points on the A^i-side. $\alpha^i_s(x)$: the index of this box. $t^i_s(x)$: the associated value expected in traces $T^i(z)$ for $z \in M^i_s(x)$. $k^i_s(x)$: the level at which x points on the A^i-side, so $|T^i(z)| \leqslant k^i_s(x)$ for all $z \in M^i_s(x)$. $\texttt{top}_s(x)$: the side, 0 or 1, from which x is expecting permission next.
- Defined from these parameters: $v^i_s(x)$, the use of reducing $E(x)$ to A^i.
- $a(k) = 2k+2$. $J(k)$: the private box at level k. $B(k,\beta)$, for $\beta \in {}^{\leqslant k+1}a(k)$: a sub-box indexed by β.
- $\beta^i_s(k)$ and $m^i_s(k)$: a string and a number such that the box $B(k, \beta^i_s(k)\hat{\ } m^i_s(k))$ is available as a new value for some $\alpha^i_s(x)$.

Construction. $s = 0$ is a *stage*; at stage 0, we do nothing except for defining $\psi^i_0(z) = 0$ for all $z < \omega$ and both $i < 2$. Let $s > 0$; let r be the last *stage* prior to s. As described above, s is a *stage* if for all z mentioned by stage r, for both $i < 2$, $\psi^i_r(z) \in T^i_s(z)$. If s is not a *stage*, then we do nothing at stage s, and we let all objects of the construction maintain their previous values; in particular, for all z and $i < 2$, $\psi^i_s(z) = \psi^i_{s-1}(z) = \psi^i_r(z)$.

Suppose that s is a *stage*. A requirement P^e *requires attention* at stage s if it is not yet satisfied, and either:

(1) every follower of P^e which is currently alive is realised (this includes the case that it has no followers); or
(2) some realised follower of P^e is permitted.

If no requirement requires attention, we do nothing, and let all objects of the construction maintain their previous values. Otherwise, we let P^e be the strongest requirement which requires attention at stage s.

If (1) above holds, we appoint a new, large follower x for P^e. We cancel all followers for requirements weaker than P^e. We set up x's parameters as follows:

- We define $k^0_s(x) = k^1_s(x) = e$.
- We let $\texttt{top}_s(x) = 1$.
- We let $M^1_s(x) = J(e)$ (so x begins its life by residing in its private box).
- By $\otimes_s$, we let $M^0_s(x) = B\left(e, \beta^0_s(e)\hat{\ } m^0_s(e)\right)$.
- We let $t^0_s(x) = t^1_s(x) = s$.

To facilitate this, we define, for both $i < 2$ and all $z \in M^i_s(x)$, $\psi^i_s(z) = s$. For all other z, we let $\psi^i_s(z) = \psi^i_{s-1}(z) = \psi^i_r(z)$, and end the stage.

If (2) above holds, let x be the strongest follower for P^e which is permitted at stage s. We cancel all followers weaker than x. Now we need to let x move; there are three cases.

A. If $k^1_{s-1}(x) = k^1_r(x)$ is undefined, this means that x has open permission from A^1, and has just received permission from A^0 (so $\texttt{top}_{s-1}(x) = 0$ and $k^0_{s-1}(x) = c$). So we now have double permission and so we enumerate x into E. As P^e now becomes satisfied, we cancel all the followers for P^e.

B. If $\mathtt{top}_{s-1}(x) = 1$ and $k^1_{s-1}(x) = c$, then A^1 now permits x, but there is no stronger A^1-box for x to be promoted to. In this case, we leave the A^1-permission open from now on. This means that $k^1_s(x)$, and so $M^1_s(x)$ and $t^1_s(x)$, are all undefined. We define $\mathtt{top}_s(x) = 0$, which means that from now we are seeking the A^0-permission which will land us in case (1). We leave $k^0_s(x) = k^0_{s-1}(x)$, $M^0_s(x) = M^0_{s-1}(x)$ and $t^0_s(x) = t^0_{s-1}(x)$.

C. If neither (A) nor (B) hold, then we can have a "regular" promotion for x. Let $i = \mathtt{top}_{s-1}(x)$.

- We let $\mathtt{top}_s(x) = 1 - i$. We leave $k^{1-i}_s(x) = k^{1-i}_s(x)$, $M^{1-i}_s(x) = M^{1-i}_{s-1}(x)$, and $t^{1-i}_s(x) = t^{1-i}_{s-1}(x)$.
- We let $k^i_s(x) = k^i_{s-1}(x) - 1$. We note that $k^i_s(x) \geqslant c$, for otherwise we would be in case (1) (if $i = 0$) or case (2) (if $i = 1$).
- By $\otimes_s$, we let $M^i_s(x) = B\left(k^i_s(x), \beta^i_s\left(k^i_s(x)\right)\hat{}m^i_s\left(k^i_s(x)\right)\right)$.
- For all $z \in M^i_s(x)$, we let $t^i_s(x) = s$.

To facilitate this, we set $\psi^i_s(z) = s$ for all $z \in M^i_s(x)$. For all other z, we set $\psi^i_s(z) = \psi^i_{s-1}(z)$. For all z, we set $\psi^{1-i}_s(z) = \psi^{1-i}_{s-1}(z)$.

In any of the cases above, we then end the stage. This completes the construction.

2.3. **Justification.** Before we verify that all requirements are met, we need to show that the construction can actually be carried out as described: we need to show that $\otimes_s$ holds at every stage s. The proof will follow a careful analysis of how metaboxes are used, allowing us to establish bounds on the number of followers processed by these boxes at any given stage. We henceforth fix a stage s^*, assume that $\otimes_s$ holds for all $s < s^*$, and so the construction is performed up to stage s^*. We show that $\otimes_{s^*}$ holds. We may assume that s^* is a *stage* .

We first establish some basic facts and notation. To begin, for $e \geqslant c$ and $s < s^*$, let F^e_s be the collection of followers for P^e which are alive at the end of stage s. We let $F_s = \bigcup_{e \geqslant c} F^e_s$ be the collection of all followers alive at the end of stage s.

For $x \in F_s$, let $R_s(x) = \{0, 1\}$ if both $k^0_s(x)$ and $k^1_s(x)$ are defined, and let $R_s(x) = \{0\}$ if only $k^0_s(x)$ is defined (and $k^1_s(x)$ is undefined).

The following is immediate from the construction.

Lemma 2.1. *Let $x \in F_s$.*

1. *For $i \in R_s(x)$, we have $k^i_s(x) \in [c, e]$.*
2. $\mathtt{top}_s(x) \in R_s(x)$.
3. *Exactly one of the following holds:*
   - $k^0_s(x) = k^1_s(x)$ *and* $\mathtt{top}_s(x) = 1$;
   - $k^0_s(x) = k^1_s(x) + 1$ *and* $\mathtt{top}_s(x) = 0$;
   - $R_s(x) = \{0\}$ *and* $k^0_s(x) = c$.
4. *If $t < s$, $x \in F_t$ and $i \in R_s(x)$, then $i \in R_t(x)$ and $k^i_s(x) \leqslant k^i_t(x)$.*
5. *Suppose that $x \in F_{s-1} \cap F_s$ and $i \in R_s(x)$. If $k^i_{s-1}(x) \neq k^i_s(x)$ then $i =$ $\mathtt{top}_{s-1}(x)$, $1 - i = \mathtt{top}_s(x)$, and $k^i_s(x) = k^i_{s-1}(x) - 1$. If $r < s$, $x \in F_r$ and $k^i_r(x) = k^i_s(x)$ then $M^i_r(x) = M^i_s(x)$ and $t^i_r(x) = t^i_s(x)$.*

In light of part (5) of Lemma 2.1, for any follower x, $k \geqslant c$ and $i < 2$, the values of $M^i_s(x)$ and $t^i_s(x)$ are constant for stages s such that $x \in F_s$, $i \in R_s(x)$ and $k^i_s(x) = k$. If there is any such stage, we let $M^i(k,x)$ and $t^i(k,x)$ be these constant values. If $M^i(k,x) \neq J(k)$, then we let $\alpha^i(k,x) = \alpha^i_s(x)$ for such a stage s be the string α such that $M^i_k(x) = B(k,\alpha)$.

We give names to sets of followers, in light of part (3) of Lemma 2.1. Let $i < 2$, and let $k \geqslant c$. Let $s < s^*$.

- We let $K^i_s(k)$ be the collection of followers $x \in F_s$ such that $i = \mathtt{top}_s(x)$, $k = k^i_s(x)$, and $M^i_s(x) = M^i(k,x)$ is not $J(k)$; that is, x does not point to its private box at the end of stage s.
- We let $L^i_s(k)$ be the collection of followers $x \in F_s$ such that $i = \mathtt{top}_s(x)$, $k = k^i_s(x)$, and $M^i_s(x) = M^i(k,x) = J(k)$. Indeed $L^i_s(k)$ is nonempty only if $i = 1$, and every $x \in L^1_s(k)$ is a follower for P^k.
- We let $G^i_s(k)$ be the collection of followers $x \in F_s$ such that $i \neq \mathtt{top}_s(x)$ but $i \in R_s(x)$ and $k^i_s(x) = k$.

For any i, k and s, the sets $K^i_s(k)$, $L^i_s(k)$ and $G^i_s(k)$ are pairwise disjoint.

For brevity, we let $KG^i_s(k) = K^i_s(k) \cup G^i_s(k)$, $LG^i_s(k) = L^i_s(k) \cup G^i_s(k)$, and so on. So:

- $KLG^i_s(k)$ is the collection of followers $x \in F_s$ such that $i \in R_s(x)$ and $k^i_s(x) = k$;
- $KL^i_s(k)$ is the collection of followers $x \in KLG^i_s(k)$ such that $i = \mathtt{top}_s(x)$; and
- $KG^i_s(k)$ is the collection of followers $x \in KLG^i_s(k)$ such that $\alpha^i_s(x)$ is defined, i.e. such that $M^i_s(x) \neq J(k)$.

For $X \in \{K, L, G, KL, KG, LG, KLG\}$, we let $\overline{X^i_s}(k) = X^i_s(k) \cap X^i_{s-1}(k)$.

The following lemma translates the construction into our new notation. In all of the following lemmas, let $i < 2$, $k \geqslant c$ and $s < s^*$ be a *stage*.

Lemma 2.2. *Let $x \in KLG^i_s(k)$. Let $t = t^i_s(x) = t^i(k,x)$. The* stage *t is the least stage r such that $x \in KLG^i_r(k)$. At stage t, x is placed into $LG^i(k)$.*

- *If x was appointed at stage t, then x is placed into $L^1_t(k)$ and $G^0_t(k)$.*
- *Otherwise, x is realised at stage t, $i = \mathtt{top}_{t-1}(x)$ (and $1-i = \mathtt{top}_t(x)$), x is extracted from $KL^i_{t-1}(k+1)$ and placed into $G^i_t(k)$; and moved from $G^{1-i}_{t-1}(k')$ to $K^{1-i}_t(k')$, where $k' = k^{1-i}_t(x)$.*

As a corollary, if $x \in KG^i_s(k)$, then x enters $G^i(k)$ at stage t, and so $M^i(k,x) = B\left(k, \beta^i_t(k)\hat{\ } m^i_t(k)\right)$. The string $\beta^i_t(k)$ is defined as follows:

- *If $\overline{K^i_t}(k) = \varnothing$ then $\beta^i_t(k) = \langle\rangle$.*
- *If $\overline{K^i_t}(k) \neq \varnothing$ then $\beta^i_t(k) = \alpha^i(k,w)$, where $w = \max \overline{K^i_t}(k)$.*

The number $m^i_t(k)$ is chosen so that for all $y \in \overline{KG^i_t}(k)$, $\alpha^i(k,y) \neq \beta^i_t(k)\hat{\ } m^i_t(k)$.

The associated values $t^i_s(x)$ obey the priority ordering on followers:

Lemma 2.3. *Let $x, y \in F_s$ with $x < y$. Let $i \in R_s(x)$ and $j \in R_s(y)$. Then $t^i_s(x) < t^j_s(y)$.*

Proof. Let r be the stage at which y is appointed; so $t^j_s(y) \geqslant r$. At stage $t^i_s(x)$, x receives attention; if $t^i_s(x) \geqslant r$ then y would be cancelled at stage $t^i_s(x)$. Hence $t^i_s(x) < r$. □

The tree of occupied boxes. Let

$$O^i_s(k) = \left\{\alpha^i(k,x) \,:\, x \in KG^i_s(k)\right\}.$$

This is the collection of indices of metaboxes which are occupied at stage s.

Lemma 2.4. *The function $x \mapsto \alpha^i(k,x)$ as x ranges over the followers in $KG^i_s(k)$ is injective.*

Proof. Let $y < x$ be elements of $KG^i_s(k)$. Let $t = t^i(k,x)$. As in the proof of Lemma 2.3, y does not receive attention between stages t and s, as this would cancel x. Hence $y \in \overline{KG^i_t}(k)$. Now by Lemma 2.2 we see that the choice of $\alpha^i(k,x)$ at stage t ensures that $\alpha^i(k,y) \neq \alpha^i(k,x)$. □

We note that $O^i_s(k)$ does not contain the empty string. This is because every $\alpha^i(k,x)$ is chosen to be $\beta^i_t(k)\hat{\ }m^i_t(k)$ for $t = t^i(k,x)$.

Lemma 2.5. *Let $x \in KG^i_s(k)$. If $\gamma \subsetneq \alpha^i(k,x)$ is nonempty, then there is some $y \in K^i_s(k)$, stronger than x, such that $\gamma = \alpha^i(k,y)$.*

Proof. Let $x \in KG^i_s(k)$, and suppose that $|\alpha^i(k,x)| > 1$; let $\gamma = \alpha^i_k(x)^-$ be the immediate predecessor of $\alpha^i(k,x)$. We show that there is some $y < x$ in $K^i_s(k)$ such that $\gamma = \alpha^i(k,y)$. Then part (2) follows by induction on $|\alpha^i(k,x)|$.

Let $t = t^i(k,x)$. At stage t we choose $\alpha^i(k,x) = \beta^i_t(k)\hat{\ }m^i_t(k)$, so $\gamma = \beta^i_t(k)$. Since $\beta^i_t(k) = \gamma \neq \langle\rangle$, we have $\overline{K^i_t}(k) \neq \varnothing$. Hence (Lemma 2.2) $\gamma = \alpha^i(k,w)$ for $w = \max \overline{K^i_t}(k)$. Since w was not cancelled at stage t, w is stronger than x. Since x is not cancelled between stages t and s, w does not receive attention between these stages, and so $w \in K^i_s(k)$. □

Lemma 2.5 implies that ordered by inclusion, $O^i_s(k)$ is a forest, and that for all $x \in G^i_s(k)$, $\alpha^i(k,x)$ is a leaf of $O^i_s(k)$. It follows that for all $x \in KG^i_s(k)$, at stage $t = t^i(k,x)$, $\alpha^i(k,x)$ is a leaf of $O^i_t(k)$, as $x \in G^i_t(k)$.

The leaves of $O^i_s(k)$ all issue from a single stem $\left\{\alpha^i(k,x) \,:\, x \in K^i_s(k)\right\}$:

Lemma 2.6. *Let $x, y \in K^i_s(k)$ with $y < x$. Then $\alpha^i(k,y) \subsetneq \alpha^i(k,x)$.*

And so $M^i(k,x) \subsetneq M^i(k,y)$.

Proof. The lemma is proved by induction on s. Let $t = t^i(k,x)$. Since y is stronger than x and $y \in K^i_s(k)$, we have $y \in \overline{K^i_t}(k)$. Hence $\overline{K^i_t}(k) \neq \varnothing$, and so $\beta^i_t(k) = \alpha^i(k,w)$ for $w = \max \overline{K^i_t}(k)$. By its definition, $y \leqslant w$. By induction, as $t-1 < t \leqslant s$, we have $\alpha^i(k,y) \subseteq \alpha^i(k,w)$. By its choice, we have $\alpha^i(k,x) \supset \beta^i_t(k) = \alpha^i(k,w)$. Hence $\alpha^i(k,y) \subsetneq \alpha^i(k,x)$. □

Let

$$\overline{O^i_s}(k) = \left\{\alpha^i(k,x) \,:\, x \in \overline{KG^i_s}(k)\right\}.$$

We can rephrase the statement $\otimes_s$ using this notation. It says that $|\beta^i_s(k)| \leqslant k$ and that there is some $m < a(k)$ such that $\beta^i_s(k)\hat{\ }m \notin \overline{O^i_s}(k)$. For in the situation described by $\otimes_s$, the followers $w \neq x$ alive at the end of stage s are precisely the followers in $\overline{F_s}$; for $w \in \overline{F_s}$ we have $k^i_s(w) = k$ and $\alpha^i_s(k)$ defined (and equals $\alpha^i(k,w)$) if and only if $w \in \overline{KG^i_s}(k)$.

Impermissiveness. We work toward finding bounds on the sizes of the sets of followers we have defined. To do so, we tie followers to values in traces. The first lemma shows that if the value $t^i(k,x)$ of a follower is not traced, then the follower x is permitted.

Lemma 2.7. *Let s be a* stage. *Let $x \in F_{s-1}$, let $i = \mathtt{top}_{s-1}(x)$ and $t = t^i_{s-1}(x)$, and suppose that there is some $z \in M^i_{s-1}(x)$ such that $t \notin T^i_s(z)$. Then x is permitted at stage s.*

Proof. Suppose that x is not permitted at stage s. Let r be the previous *stage* before stage s; so $x \in F_r$, $i = \mathtt{top}_r(x)$ and $t = t^i_r(x)$. Since x is not permitted at stage s, an i-computation is defined for x at stage r, and $A^i_s \upharpoonright v^i_r(x) = A^i_s \upharpoonright v^i_r(x)$. Let $z \in M^i_{s-1}(x) = M^i_r(x)$. Then $t = t^i_r(x) \in T^i_r(x)$, and by its definition, $v^i_r(x) \geqslant u^i_r(z,t)$, the A^i_r-use of enumerating t into $T^i_r(z)$. The fact that A^i did not change below $v^i_r(x)$ between stages r and s shows that $t \in T^i_s(x)$ as well. □

The second lemma says that we do not change ψ^i too often. This will also be useful in the verification, to show that the use of reducing E to A^i does not go to infinity.

Lemma 2.8. *Suppose that either*

- *$x \in G^i_s(k)$, or*
- *$x \in L^i_s(k)$, and x is unrealised at stage s.*

Then for all $z \in M^i(k,x)$, $\psi^i_s(z) = t^i(k,x)$.

Proof. Let $z \in M^i(k,x)$ (in the second case, z is the unique element of $M^i(k,x)$). At stage $t = t^i(k,x)$, we set $\psi^i_t(z) = t$. We need to show that at no stage $r \in (t,s]$ do we redefine the value of $\psi^i(z)$ to be r.

In the second case, this follows from the fact that as $z = \min I(k)$ is the unique element of $J(k)$, we only define $\psi^i_r(z) = r$ at a stage r if at that stage, the requirement P^k appoints new follower. However, the assumption that x is unrealised at stage s implies that x is unrealised at all stages $r \in [t,s]$, and the instructions tell P^k to appoint a new follower only when all of its followers are realised. So at no stage $t \in (t,s]$ does P^k appoint a new follower, and so $\psi^i(z)$ is unchanged between stages t and s.

In the first case, suppose, for contradiction, that at stage $r \in (t,s]$ we redefine $\psi^i_r(z) = r$. Since $z \in I(k)\backslash J(k)$, by the instructions, there is some follower y which enters $G^i(k)$ at stage r, and $z \in M^i(k,y)$. By Lemma 2.5, $\alpha^i(k,y)$ is a leaf of $O^i_r(k)$. Since $x \in G^i_s(k)$, we must have $x \in G^i_r(k)$, and so by the same lemma, $\alpha^i(k,x)$ is also a leaf of $O^i_r(k)$. Since x does not enter $G^i(k)$ at stage r (as $r > t$), we have $x \neq y$. Hence $\alpha^i(k,x)$ and $\alpha^i(k,y)$ are incomparable, which means that $M^i(k,x)$ and $M^i(k,y)$ are disjoint, contradicting $z \in M^i(k,x) \cap M^i(k,y)$. □

Lemma 2.9. *Let s be a* stage. *For all $x \in \overline{KLG^i_s}(k)$, for all $z \in M^i(k,x)$, $t^i(k,x) \in T^i_s(z)$.*

So $v^i_s(x)\downarrow$.

Proof. There are three possibilities: either $x \in K^i_{s-1}(k)$, or $x \in G^i_{s-1}(k)$, or $x \in L^i_{s-1}(k)$.

If $x \in K^i_{s-1}(k)$ (so $i = \mathtt{top}_{s-1}(x)$), then the fact that $x \in K^i_s(k)$ means that x is not permitted at stage s. For otherwise, either x is cancelled at stage s by some stronger follower, of x receives attention at stage s, in which case x is extracted at stage s from $K^i(k)$ and possibly moved to $K^i_s(k-1)$. In this case, the conclusion follows from Lemma 2.7.

Now suppose that $x \in G^i_{s-1}(k)$. Let $z \in M^i(k,x)$. By Lemma 2.8, $\psi^i_{s-1}(z) = t^i(k,x)$. The conclusion then follows from the fact that s is a *stage*: let r be the previous *stage* prior to s. As $z \in M^i_{s-1}(x) = M^i_r(x)$, z is mentioned by stage r, and so $\psi^i_{s-1}(z) = \psi^i_r(z) \in T^i_s(z)$.

Finally, suppose that $x \in L^i_s(k)$ (so $i = 1$). There are two sub-cases. If x is realised at stage s, then by the fact that $x \in L^i_s(k)$ it follows that x is not permitted at stage s. Then the conclusion follows from Lemma 2.7.

We assume, then, that x is unrealised at stage s. Let z be the unique element of $M^i(k,x) = J(k)$. By Lemma 2.8, $\psi^i_{s-1}(z) = t^i(k,x)$. As in the second case above, the conclusion follows from the fact that s is a *stage*. ☐

Sizes of sets of followers. We can now bound the sets of followers.

Lemma 2.10. *For all $s < s^*$, $\left|K^i_s(k)\right| \leqslant k$.*

Proof. Since $K^i(k)$ does not change between *stages*, we may assume that s is a *stage*. Assuming that $K^i_s(k)$ is nonempty, let $w = \max K^i_s(k)$. By Lemma 2.6, for all $x \in K^i_s(k)$, $M^i(k,w) \subseteq M^i(k,x)$. Let z be any element of $M^i(k,w)$, and let $x \in K^i_s(k)$. Then $x \in \overline{GK^i_s}(k)$ (Lemma 2.2). By Lemma 2.9, $t^i(k,x) \in T^i_s(z)$. By Lemma 2.3, the map $x \mapsto t^i(k,x)$ is injective on $K^i_s(k)$. The conclusion follows from the fact that $|T^i_s(z)| \leqslant k$ as $z \in I(k)$ (so $h(z) = k$). ☐

Lemma 2.10 implies the first part of $\otimes_{s^*}$.

Lemma 2.11. $|\beta^i_{s^*}(k)| \leqslant k$.

Proof. If $\beta^i_s(k) = \langle\rangle$ we are done. Otherwise, $\beta^i_{s^*}(k) = \alpha^i_{s^*-1}(k)$ for some $w \in K^i_{s^*-1}(k)$. By Lemma 2.5, $|\alpha^i_{s^*-1}(k)| \leqslant |K^i_{s^*-1}(k)|$. The result follows from Lemma 2.10. ☐

Lemma 2.12. *For all $s < s^*$, $\left|L^i_s(k)\right| \leqslant k+1$.*

Of course $L^0_s(k)$ is always empty, so this is of interest for $i = 1$.

Proof. Again we may assume that s is a *stage*. Let $z = \min I(k)$ be the unique element of $J(k)$; so $h(z) = k$, which means that $|T^i_s(z)| \leqslant k$. By Lemma 2.9, for all $x \in \overline{L^i_s(k)}$, $t^i(k,x) \in T^i_s(z)$; with Lemma 2.3 we see that $\left|\overline{L^i_s}(k)\right| \leqslant k$. At stage s, at most one follower is added to $L^i(k)$. ☐

So we see that $|KL^i_s(k)| \leqslant 2k+1$. Recall that $a(k) = 2k+2$.

Lemma 2.13. *For all $s < s^*$, $\left|G^i_s(k)\right| < a(k)$.*

Proof. Let $x \in G^i_s(k)$. By Lemma 2.2, if $i = 0$ then $x \in LK^1_s(k)$, and if $i = 1$, then $x \in K^0_s(k+1)$. Hence $|G^0_s(k)| \leqslant 2k+1 < a(k)$ and $|G^1_s(k)| \leqslant k+1 < a(k)$. ☐

We can finally prove $\otimes_{s^*}$.

Proof of $\otimes_{s*}$*:* The first part is Lemma 2.11. It remains to show that there is some $m < a(k)$ such that $\beta^i_{s*}(k)\hat{\ }m \notin \overline{O^i_{s*}}(k)$. We observe that for all $m < a(k)$, if $\beta^i_{s*}(k)\hat{\ }m \in \overline{O^i_{s*}}(k)$, then $\beta^i_{s*}(k)\hat{\ }m = \alpha^i(k,x)$ for some $x \in G^i_{s-1}(k)$. This would imply the lemma, using the bound on the size of $G^i_{s*-1}(k)$ given by Lemma 2.13. To see that $\beta^i_{s*}(k)\hat{\ }m$ cannot be $\alpha^i(k,x)$ for $x \in K^i_{s-1}(k)$, suppose for contradiction that it is; then $x \in K^i_{s*}(k)$ (as it is in $KG^i_{s*}(k)$), and so $\overline{K^i_{s*}}(k)$ is nonempty, whence $\beta^i_{s*}(k) = \alpha^i(k,w)$ for $w = \max \overline{K^i_s}(k)$. But then $x \leqslant w$ and so (Lemma 2.6) we would have $\alpha^i(k,x) \subseteq \alpha^i(k,w)$ for a contradiction. □

2.4. **Verification.** Having shown that the construction can proceed as described, we now show that it succeeds in enumerating a set with the desired properties.

Lemma 2.14. *There are infinitely many* stages*.*

Proof. Indicated above. Let s be a *stage*, and suppose for a contradiction that there is no greater *stage*. Then for both $i < 2$, $\psi^i = \psi^i_s$. For every input z mentioned by stage s, $\psi_i(z) \in T^i(z)$. There is a sufficiently large stage $t > s$ such that for all such z (as there are only finitely many of them), $\psi_i(z) \in T^i_s(z)$ by an A^i-correct computation. Hence there must be a *stage* greater than s after all, contradiction. □

Fairness and diagonalisation.

Lemma 2.15. *Every follower receives attention only finitely many times.*

Proof. For $x \in F_s$, let $k_s(x) = \sum_{i \in R_s(x)} k^i_s(x)$. If $x \in F_{s-1} \cap F_s$ then $k_s(x) \leqslant k_{s-1}(x)$ (Lemma 2.2).

Suppose that $x \in F_{s-1} \cap F_s$ is a follower which receives attention at stage s. Then either $k^0_s(x) < k^0_{s-1}(x)$ or $k^1_s(x) < k^1_{s-1}(x)$. Hence $k_s(x) < k_{s-1}(x)$. This can happen at most finitely many times. Indeed, after being appointed, each follower for requirement P^e can receive attention at most $2e$ many times. □

For $i < 2$, $k \geqslant c$ and $X \in \{K, L, G\}$, let $X^i_\omega(k) = \lim_s X^i_s(k)$ be the collection of followers x which are in $X^i_s(k)$ for all but finitely many s. Lemma 2.10 shows that $|K^i_\omega(k)| \leqslant k$ and Lemma 2.12 shows that $|L^i_\omega(k)| \leqslant k+1$ (in fact, the proof of Lemma 2.12 shows that $|L^i_\omega(k)| \leqslant k$; but from now, we only care that it is finite).

Lemma 2.16. *For every $e \geqslant c$, the requirement P^e is met, and there is some stage after which no follower for P^e ever requires attention. In particular, eventually P^e stops choosing new followers.*

Proof. By induction on $e \geqslant c$. Suppose this has been verified for all $e' < e$.

Of course, if P^e is ever satisfied, then it is met, and ceases all action. We assume, then, that no follower for P^e is ever enumerated into E.

Let H^e be the collection of followers x for P^e which are never cancelled. Then by Lemma 2.15,

$$H^e \subseteq L^1_\omega(e) \cup \bigcup_{k \in [c,e], i<2} K^i_\omega(k).$$

This shows that H^e is finite.

Let s_0 be the last stage at which any follower for a requirement stronger than P^e receives attention ($s_0 = 0$ if there are no such followers). At stage s_0, all followers for P^e are cancelled. At the next *stage* after stage s_0, P^e will appoint a new follower. This follower can never be cancelled. This shows that H^e is nonempty.

Let s_1 be the last stage at which any follower in H^e receives attention. First we note that $H^e = F^e_{s_1}$, the set of followers for P^e which are alive at the end of stage s_1. For if $w \in H^e$ receives attention at stage s_1, and $x \in F^e_{s_1}$, then $x \leqslant w$; since w is not cancelled at a later stage, neither is x.

We claim that P^e does not appoint new followers after stage s_1. For if it does, let s be the least stage greater than s_1 at which P^e appoints a follower x. This follower x is stronger than any other follower for P^e (at any stage $t \geqslant s$) other than the followers in H^e. As the followers in H^e do not require attention after stage s_1, x can never be cancelled. But this means that $x \in H^e$, a contradiction with the maximality of s_1.

The fact that P^e does not appoint any followers after stage s_1 implies that no followers for P^e require attention after stage s_1; so P^e's overall action is finitary. It also implies that the weakest follower $w \in H^e$ is never realised. For if it is, then all followers in H^e are realised at some point, and P^e would be instructed to appoint a new follower. As $x \notin E$, this shows of course that $E \neq \varphi^e$, and so P^e is met. □

It follows that E is not computable.

Reductions.

Lemma 2.17. *Let s be a* stage*, and let $x \in F_s$. Let $i \in R_s(x)$, and suppose that $v^i_s(x)\downarrow$. Suppose that $A^i \upharpoonright v^i_s(x) = A^i_s \upharpoonright v^i_s(x)$. Then $x \notin E$.*

Proof. By induction on $r > s$ we can see that if $x \in F_s$ then $i \in R_r(x)$ and $k^i_r(x) = k^i_s(x)$, and so that $t^i_r(x) = t^i_s(x)$; that $v^i_r(x)\downarrow = v^i_s(x)$, and that if $i = \mathtt{top}_{r-1}(x)$ then x is not permitted at stage r.

The point is that if $x \in E$ then there must be some stage $r \geqslant s$ such that $i = \mathtt{top}_r(x)$; if $i \neq \mathtt{top}_s(x)$, then $i = \mathtt{top}_r(x)$ where r is the next stage at which x receives attention. But then, the fact that x will not be permitted once i becomes $\mathtt{top}(x)$, means that x cannot be enumerated into E.

This, of course, is where we use the fact that A^i is c.e., rather than merely Δ^0_2. □

The next lemma states that unless cancelled, enumerated into E, or given open permission from A^1, the use $v^i(x)$ of reducing the statement $x \notin E$ to A^i stabilizes. Using the notation above, let H be the collection of followers which are never cancelled nor enumerated into E. For $x \in H$, let $R_\omega(x) = \lim_s R_s(x)$ be the collection of $i < 2$ such that $i \in R_s(x)$ for all but finitely many s. For $i \in R_\omega(x)$, let $k^i_\omega(x) = \lim_s k^i_s(x)$, and so on.

Lemma 2.18. *Let $x \in H$ and let $i \in R_\omega(x)$. There is some* stage *s such that $x \in F_s$, $v^i_s(x)\downarrow$ and $A^i \upharpoonright v^i_s(x) = A^i_s \upharpoonright v^i_s(x)$.*

Proof. Let $k = k^i_\omega(x)$ and let $t = t^i_\omega(x) = t^i(k, x)$. There are two cases.

If either $x \in G^i_\omega(k)$, or $x \in L^i_\omega(k)$ and x is never realised, then Lemma 2.8 implies that for all $z \in M^i(k, x)$, $\psi^i(z) = t$. As T^i traces ψ^i, we have $t \in T^i(z)$ for all $z \in M^i(k, x)$. There is a *stage* $s > t$ at which for all $z \in M^i(k, x)$, $t \in T^i_s(z)$ by an A^i-correct computation. Then s is as required.

In the second case, $x \in KL^i_w(k)$ and x is eventually realised. Let $s > t$ be a *stage* such that $x \in \overline{KL^i_s}(k)$. By Lemma 2.9, $v^i_s(x)\downarrow$. We claim that s is a stage as required by the lemma. For if not, there is a least *stage* $r > s$ by which we see an A^i-change below $v^i_s(x)$. Since by induction $i = \mathtt{top}_{r-1}(x)$, at stage r, x would be permitted, contradicting the definition of k. ☐

Lemma 2.19. $E \leqslant_{\mathrm{T}} A^0$.

The point is that for $x \in H$ we always have $0 \in R_\omega(x)$.

Proof. Let $x < \omega$. To decide, with oracle A^0, whether $x \in E$ or not, first see if $x \in F_x$. If not, then $x \in E$ if and only if $x \in E_x$.

Suppose that $x \in F_x$. By Lemmas 2.15 and 2.18, with oracle A^0, we can find a *stage* $s > x$ at which one of the following hold:

- $x \notin F_s$ (that is, x has been cancelled by stage s).
- $x \in E_s$ (that is, x has been enumerated into E by stage s).
- $v^0_s(x)\downarrow$, and $A^0 \restriction v^0_s(x) = A^0_s \restriction v^0_s(x)$.

By Lemma 2.17, this allows A^0 to decide whether $x \in E$ or not. ☐

Recall that the difference between A^0 and A^1 is that there may be $x \in H$ such that $1 \notin R_\omega(x)$: these are the followers that receive open permission from A^1 but do not get later permission from A^0.

Lemma 2.20. *There are only finitely many followers $x \in H$ such that $1 \notin R_\omega(x)$.*

Proof. Suppose that $x \in H$ and $1 \notin R_\omega(x)$. Then $x \in K^0_\omega(c)$. By Lemma 2.10, $|K^0_\omega(c)| \leqslant c$. ☐

Lemma 2.21. $E \leqslant_{\mathrm{T}} A^1$.

Proof. Similar to the proof of Lemma 2.19. For a follower $x \in F_x$, we search for a *stage* $s > x$ by which it is either cancelled, enumerated, or $v^1_s(x)\downarrow$ and A^1_s is correct up to this value. Lemma 2.20 says that this search will terminate for all but finitely many followers x, and so non-uniformly will give a method for reducing E to A^1. ☐

3. Proof of Theorem 1.8

In this section we adapt the construction of the previous section and provide a construction of a noncomputable c.e. set E, computable from every SJT-hard c.e. set, thus proving Theorem 1.8.

3.1. **Discussion.** There is only one really new ingredient, and our treatment is not too surprising to those familiar with Π^0_2 constructions on trees. Instead of being given two (or finitely many) SJT-hard c.e. sets, together with traces for Σ^0_2 functions we approximate, we need to guess, among all pairs of c.e. sets and possible traces, which indeed trace the functions that we enumerate. The construction is performed on a tree of strategies. Nodes τ on the tree will test if there are infinitely many τ-*stages*, at which we can calculate uses of reducing $x \in E$ to the corresponding c.e. set W^e.

The small degree of non-uniformity which was necessary in the construction of the previous section plays an important role. A follower x for some node σ on the tree can be cleared by only finitely many nodes τ for which σ guesses that there are infinitely many τ-*stages*. In other words, it requires eventual permission from only finitely many c.e. sets W^e. Other SJT-hard sets do not comprehend x's existence.

This is akin to those sets giving x immediate open permission. The tree machinery ensures that each such set is troubled by at most finitely many such followers.

3.2. **Construction.** As before, we enumerate a set E. To ensure that E is non-computable, we meet the same positive requirements P^e as in the previous section, which state that $E \neq \varphi^e$, where $\langle \varphi^e \rangle_{e<\omega}$ is a list of all partial computable functions.

Let $\langle W^e \rangle_{e<\omega}$ be an effective list of all c.e. sets. Shortly we will define, for all $e < \omega$, an order function h^e. Let, uniformly in e, $\langle T^{e,c} \rangle_{c<\omega}$ list all W^e-c.e. traces which are bounded by h^e. During the construction we define, uniformly in e, a partial Σ^0_2 function ψ^e. The negative requirements are named $N^{e,c}$, and state that if $T^{e,c} = \langle T^{e,c}(z) \rangle_{z<\omega}$ traces ψ^e, then E is computable from W^e.

The construction takes place on a tree of strategies. The definition of the tree is recursive: given a node (a strategy) on the tree, the immediate successors of the node on the tree are determined by the possible outcomes of the node. If a node σ works for P^e, then it has a single outcome. If a node τ works for $N^{e,c}$, then τ has two outcomes, ∞ and `fin`, which indicate, roughly, whether $T^{e,c}$ is a trace for ψ^e or not. The outcome ∞ is stronger than the outcome `fin`, and this ordering induces a total priority ordering on the tree. The outcome ∞ indicates that there are infinitely many τ-*stages*.

Let $\mathcal{P}^e$ be the collection of nodes on the tree that work for P^e, and $\mathcal{N}^{e,c}$ be the collection of nodes that work for $N^{e,c}$. We let $\mathcal{N}^e = \bigcup_c \mathcal{N}^{e,c}$, $\mathcal{N} = \bigcup_e \mathcal{N}^e$, and $\mathcal{P} = \bigcup_e \mathcal{P}^e$.

To complete the definition of the tree of strategies, we need to show how to assign requirements to nodes. We could simply assign each level of the tree a single requirement. However, for simplicity of presentation, we would like to assume that for all $\sigma \in \mathcal{P}$ there is some $\tau \in \mathcal{N}$ such that $\tau\hat{\ }\infty \subseteq \sigma$. The easiest way to achieve this is by recursively assigning requirements to nodes during the definition of the tree; to each node ρ we assign the strongest requirement (from an ω-list of all requirements) which has not yet been assigned to any proper initial segment of ρ, subject to the restriction that if there is no τ, which has been already placed in $\mathcal{N}$, such that $\tau\hat{\ }\infty \subseteq \rho$, then we must assign a negative requirement to ρ. After verifying that there is a true path, we will easily see that the true path contains a node of the form $\tau\hat{\ }\infty$ for some $\tau \in \mathcal{N}$ (as there are SJT-hard c.e. sets), and this would allow us to show that every requirement is assigned to some node on the true path.

The order functions. The next order of business is defining the order functions h^e. These derive from the structure of the tree and the intended structure of the boxes. The idea is that a follower x for a node $\sigma \in \mathcal{P}$ needs to be cleared by all $\tau \in \mathcal{N}$ such that $\tau\hat{\ }\infty \subseteq \sigma$: if $\tau \in \mathcal{N}^e$ then W^e-permission is required. The guess by σ that there are infinitely many τ-*stages* allows for the machinery of the previous section to operate smoothly.

The search over all traces $T^{e,c}$ (for $c < \omega$) means that the various nodes $\tau \in \mathcal{N}^e$ have to cooperate in defining a single function ψ^e: each τ gets its own column to play with. A node τ will require a number of k-boxes for various k; as there are infinitely many τ's in $\mathcal{N}^e$, to keep h^e well-defined, the smallest k such that τ requests k-boxes needs to increase with τ. For convenience of notation, we let each τ request k-boxes for $k \geqslant |\tau|$.

For a fixed τ and $k \geqslant |\tau|$, how many k-boxes? We need to count the number of possible followers that progress down the chain of boxes, similarly to what has been done in the justification of the previous section. In other words, we need to calculate bounds on the sizes of sets $G^\tau_s(k)$, $K^\tau_s(k)$ and $L^\tau_s(k)$ which are the analogues of the sets $G^i_s(k)$, $K^i_s(k)$ and $L^i_s(k)$ of the previous section. We will still have $|K^\tau_s(k)| \leqslant k$ as this is bounded by the potential size of the trace. However, it is no longer true that every $x \in G^\tau_s(k)$ is an element of the same $K^\rho_s(k')$ (for $k' \in \{k, k+1\}$): more than two sets mean that $\mathtt{top}_s(x)$ may have value among a number of nodes ρ of length at most $k' \leqslant k+1$. The number of these strings is bounded by the number of nodes of length $k+1$; as the tree is at most binary branching, the number of such nodes is bounded by 2^{k+1}.

We also need to bound $L^\rho_s(k)$ for such strings ρ; to avoid too meticulous an examination of the way requirements are assigned to nodes, we allow more than one node σ require a private k-box from ρ. In general, $\sigma \in \mathcal{P}$ will require a private $|\sigma|$-box from the longest $\tau \in \mathcal{N}$ such that $\tau\hat{\ }\infty \subseteq \sigma$. So for $\rho \in \mathcal{N}$ and $k > |\rho|$, let $\Theta^\rho(k)$ be the collection of nodes $\sigma \in \mathcal{P}$ of length k for which ρ is the longest string in $\mathcal{N}$ such that σ extends $\rho\hat{\ }\infty$; of course $|\Theta^\rho(k)| \leqslant 2^k$. Counting all these contributions, we let, for all $k < \omega$,

$$a(k) = 1 + 2^{k+2}(k+2)(1+2^{k+1}),$$

which will be a bound on $G^\tau_s(k)$. Then any $\tau \in \mathcal{N}$ will require $|\Theta^\tau(k)| + (a(k))^{k+1} \leqslant 2^k + (a(k))^{k+1}$ many k-boxes for $k \geqslant |\tau|$.

We can now define the order functions h^e. Partition ω into ω many columns $\omega^{[\tau]}$, indexed by $\tau \in \mathcal{N}^e$. For each $\tau \in \mathcal{N}^e$, partition $\omega^{[\tau]}$ into intervals $I^\tau(k)$ for $k \geqslant |\tau|$, such that for all $k \geqslant |\tau|$,

$$|I^\tau(k)| = 2^k + a(k)^{k+1}.$$

Let h^e be an order function such that for all $\tau \in \mathcal{N}^e$, for all $k \geqslant |\tau|$, for all $x \in I^\tau(k)$, $h^e(x) \leqslant k$. The fact that for all k, there are only finitely many nodes $\tau \in \mathcal{N}^e$ of length at most k, implies that such an order function can be found, and in fact defined effectively given e.

Local traces and stages. The nodes $\tau \in \mathcal{N}^e$ collaborate in defining the function ψ^e. We let $\psi^\tau = \psi^e \restriction \omega^{[\tau]}$; the node τ is responsible for defining ψ^τ. The function ψ^τ is defined to be the partial limit of a uniformly computable sequence $\langle \psi^\tau_s \rangle_{s<\omega}$.

For $\tau \in \mathcal{N}$, we will shortly define the collection of *τ-stages*. To define ψ^τ_s, we start with $\psi^\tau_0(z) = 0$ for all $z \in \omega^{[\tau]}$. If $s > 0$ is not a *τ-stage*, then we let $\psi^\tau_s = \psi^\tau_{s-1}$. Thus if there is a last *τ-stage* s, then $\psi^\tau = \psi^\tau_s$. If s is a *τ-stage*, then we may redefine $\psi^\tau_s(z) = s$ for some $z \in \omega^{[\tau]}$; for all other z, we let $\psi^\tau_s(z) = \psi^\tau_{s-1}(z)$.

Let $\tau \in \mathcal{N}^{e,c}$. For brevity, we let $T^\tau = T^{e,c} \restriction \omega^{[\tau]}$. Thus if $T^{e,c}$ is a trace for ψ^e, then T^τ is a trace for ψ^τ. For $z \in \omega^{[\tau]}$ and $s < \omega$, we let $T^\tau_s(z)$ be the collection of numbers enumerated into $T^\tau(z)$ by stage s with oracle W^e_s. We may assume that for all z and s, $|T^\tau_s(z)| \leqslant h^e(z)$, so for all $k \geqslant |\tau|$, for all $z \in I^\tau(k)$, $|T^\tau_s(z)| \leqslant k$. For $t \in T^\tau_s(z)$, we let $u^\tau_s(z,t)$ be the W^e_s-use of enumerating t into $T^\tau_s(z)$.

The collection of *τ-stages* depends on whether τ is *accessible* at stage s, a notion which we define later. Given this, we define the collection of *τ-stages*. For all τ, 0 is a *τ-stage*. Let $s > 0$, and let $\tau \in \mathcal{N}$. If τ is not accessible at stage s, then s is not a *τ-stage*. Suppose that τ is accessible at stage s; let r be the previous *τ-stage*.

Then s is a *τ-stage* if for every $z \in \omega^{[\tau]}$ mentioned by stage r, $\psi_r^\tau(z) = \psi_{s-1}^\tau(z)$ is an element of $T_s^\tau(z)$.

Followers. Nodes $\sigma \in \mathcal{P}$ appoint followers. For any follower x, we let $\sigma(x)$ be the node which appointed x. A follower x for $\sigma \in \mathcal{P}^e$ is *realised* at stage s if $\varphi_s^e(x) = 0$. The requirement P^e is *satisfied* at stage s if there is some $x \in E_s$ such that $\varphi_s^e(x) = 0$. If the requirement P^e is satisfied at stage s, then no node $\sigma \in \mathcal{P}^e$ takes any action at stage s.

With any follower x, alive at the end of a stage s, we associate auxiliary objects.

- We attach a nonempty set of nodes $R_s(x) \subset \mathcal{N}$. For all $\tau \in R_s(x)$, $\sigma(x)$ extends $\tau\hat{\ }\infty$. The set $R_s(x)$ is the set of nodes which need to clear x before it is enumerated into E.
- We define a node $\mathtt{top}_s(x) \in R_s(x)$. This is the node from which x next requires permission.
- For all $\tau \in R_s(x)$, we define a number $k_s^\tau(x) \geqslant |\tau|$. This is the level at which x points.
- For all $\tau \in R_s(x)$, we define a box $M_s^\tau(x) \subset I^\tau(k_s^\tau(x))$.
- For all $\tau \in R_s(x)$, we define a value $t_s^\tau(x) < \omega$.

Suppose that a follower x is alive at the end of stage s, and let $\tau \in R_s(x)$. Suppose that s is a *τ-stage*. We say that a τ-computation is defined for x at stage s if for all $z \in M_s^\tau(x)$ we have $t_s^\tau(x) \in T_s^\tau(z)$. In this case, we let

$$v_s^\tau(x) = \max\left\{u_s^\tau\left(z, t_s^\tau(x)\right) \,:\, z \in M_s^\tau(x)\right\}.$$

We denote the fact that a τ-computation is defined for x at stage s by writing $v_s^\tau(x)\!\downarrow$; otherwise, we write $v_s^\tau(x)\!\uparrow$.

Let s be a stage at the beginning of which a follower x is alive. Let $\tau = \mathtt{top}_{s-1}(x)$, and suppose that s is a *τ-stage*. Let r be the previous *τ-stage* before s. We say that x is *permitted* at stage s if $v_r^i(x)\!\uparrow$ or if $W_s^e \restriction v_r^\tau(x) \neq W_r^e \restriction v_r^\tau(x)$, where $\tau \in \mathcal{N}^e$. Note again that a follower x is permitted at stage s only if s is a $\mathtt{top}_{s-1}(x)$-*stage*; but x may be permitted at a stage s at which $\sigma(x)$ is not accessible: recall that $\sigma(x)$ properly extends the node $\mathtt{top}_{s-1}(x)$.

All followers alive at stage s are linearly ordered by priority, which is given to followers appointed earlier; again, at most one new follower is appointed at each stage. New followers are chosen large, and so the priority ordering coincides with the natural ordering on natural numbers.

The general structure of the stage is as follows. During stage s we inductively define the collection of nodes which are accessible at that stage. Once a node $\sigma \in \mathcal{P}$ has been declared accessible, we see if it wants to appoint a new follower or not; if it does, it ends the stage, and cancels followers for all weaker nodes. Otherwise, it lets its only child be accessible. Once a node $\tau \in \mathcal{N}$ has been declared accessible, we decide if s is a *τ-stage* or not. If not, and $|\tau| < s$, then we let $\tau\hat{\ }\mathtt{fin}$ be next accessible. If $|\tau| = s$ then we end the stage. If s is a *τ-stage*, then we see if τ permits any realised follower x. If so, then the strongest such follower x receives attention and is moved; all followers weaker than x are cancelled, and the stage is ended. If no follower is permitted, then $\tau\hat{\ }\infty$ is next accessible.

We explain why it is important that if s is a *τ-stage*, and τ permits a realised follower x at stage s, then we let x move, even if $\sigma(x)$ is not accessible at stage s. For τ permitting x means that $t^\tau(x)$ is extracted from $T^\tau(z)$ for some $z \in M_s^\tau(x)$. The set W^e (where $\tau \in \mathcal{N}^e$) needs to decide *now* whether to assign a new use for reducing $x \in E$ to W^e using the same boxes, or promote x. If x is not allowed to move, then this incident may repeat indefinitely, meaning that the use goes to infinity. After all, W^e does not know if $\sigma(x)$ will ever be accessible again.

Of course, if x is permitted but is unrealised, then it is pointing at its private box, which again will be a singleton, denoted $\{z^\tau(\sigma(x))\}$. No weaker follower changes the value of $\psi^\tau(z)$, and so in this case τ can appoint a new use based on the same value of $\psi^\tau(z)$, which will stabilise the use. Once a follower is in $K^\tau(k)$ and not in $L^\tau(k)$ (it points at public boxes), a weaker follower y such that $M^\tau(k,y) \subset M^\tau(k,x)$ may force a change in the values of $\psi^\tau(z)$ for some $z \in M^\tau(k,x)$, which lands τ in the position of having to promote x if it is permitted.

Carving the boxes. We again carve up the intervals $I^\tau(k)$ into public and private boxes, and generate a tree of public boxes. We will need more than one private box – one for each $\sigma \in \Theta^\tau(k)$. For all $\tau \in \mathcal{N}$ and $k \geqslant |\tau|$, let $J^\tau(k)$ be a subset of $I^\tau(k)$ of size 2^k, and let $B^\tau(k,\langle\rangle) = I^\tau(k)\backslash J^\tau(k)$; so $|B^\tau(k,\langle\rangle)| = a(k)^{k+1}$. As in the previous section, we recursively define $B^\tau(k,\alpha)$ for $\alpha \in {}^{\leqslant k+1}a(k)$ by starting with $B^\tau(k,\langle\rangle)$ and splitting $B^\tau(k,\alpha)$, which inductively has size $a(k)^{k+1-|\alpha|}$, into $a(k)$-many subsets $B^\tau(k,\alpha\hat{\ }m)$ (for $m < a(k)$) of equal size $a(k)^{k-|\alpha|}$. Again, for $\alpha,\beta \in {}^{\leqslant k+1}a(k)$, if $\alpha \subseteq \beta$ then $B^\tau(k,\alpha) \supseteq B^\tau(k,\beta)$, and if $\alpha \perp \beta$ then $B^\tau(k,\alpha) \cap B^\tau(k,\beta) = \varnothing$. For all $\alpha \in {}^{\leqslant k+1}a(k)$, $B^\tau(k,\alpha) \cap J^\tau(k) = \varnothing$.

Recall that $\Theta^\tau(k)$ is a set (possibly empty) of nodes $\sigma \in \mathcal{P}$ of length k; so $|\Theta^\tau(k)| \leqslant 2^k = |J^\tau(k)|$. We fix an injection $\sigma \mapsto z^\tau(\sigma)$ of $\Theta^\tau(k)$ into $J^\tau(k)$.

Let $\tau \in \mathcal{N}$. For any follower x, alive at the end of stage s, such that $\tau \in R_s(x)$, either

- $M^\tau_s(x) = \{z^\tau(\sigma(x))\}$ (x points to σ's private box at stage s); in this case $\tau = \mathtt{top}_s(x)$ and s has not moved since it was appointed; or
- $M^\tau_s(x) = B^\tau(k,\alpha)$ for some string $\alpha \in {}^{\leqslant k+1}a(k)$; we denote this string by $\alpha^\tau_s(x)$.

We proceed as before, with finding how a follower would find a new $\alpha^\tau_s(x)$. Again suppose that s is a τ-*stage*, and that we need to define some $\alpha^\tau_s(x)$ at some intended level $k = k^\tau_s(x)$. Recall that at stage s we cancel all followers weaker than x, while followers stronger than x do not move.

We define the string $\beta^\tau_s(k)$:

- If there is a follower $y < x$, alive at the beginning of stage s, such that $\mathtt{top}_{s-1}(y) = \tau$, $k^\tau_{s-1}(y) = k$, and $M^\tau_{s-1}(y) \not\subseteq J^\tau(k)$ (in the notation of the justifications, $y \in \overline{K^\tau_s}(k)$), then we let $\beta^\tau_s(k) = \alpha^\tau_{s-1}(y)$ for the weakest such follower y.
- If there is no such follower y, then we let $\beta^\tau_s(k) = \langle\rangle$.

And as before we find a number $m = m^\tau_s(k) < a(k)$ such that there is no follower $w < x$, alive at the beginning of stage s, with $k^\tau_s(w) = k$ and $\alpha^\tau_s(w) = \beta^\tau_s(k)\hat{\ }m$. Again, we formalise the statement that a choice for $\alpha^\tau_s(x)$ is possible, and presume it holds at stage s when we perform the stage:

$\otimes_s$: $|\beta^\tau_s(k)| \leqslant k$, and a number $m^\tau_s(k)$ as above exists.

Construction. At stage $s > 0$ we recursively define the finite path of nodes which are accessible at stage s. The root $\langle\rangle$ is always accessible.

First, suppose that a node $\tau \in \mathcal{N}$ is accessible at stage s. We check to see if s is a τ-*stage* or not, as described above. If not, and $|\tau| < s$, then we let $\tau\hat{\ }\mathtt{fin}$ be accessible next; if $|\tau| = s$ then we end the stage, and cancel all followers for nodes that lie to the right of τ. If s is a τ-*stage*, but there is no realised follower which τ permits at stage s, then we let $\tau\hat{}\infty$ be accessible next (unless $|\tau| = s$, when we end the stage and cancel followers for nodes that lie to the right of τ). Suppose, then, that τ permits realised followers at stage s. Let x be the strongest realised follower which is permitted by τ at stage s. We cancel all followers weaker than x. We then promote x as follows:

A. If $R_{s-1}(x) = \{\tau\}$, this means that only τ cares about x; since τ has just permitted x, we enumerate x into E. If $\sigma(x) \in \mathcal{P}^e$ then the requirement P^e is now satisfied, so we cancel all followers for $\sigma(x)$.

B. If τ is not the only element of $R_{s-1}(x)$, but $k^\tau_{s-1}(x) = |\tau|$, then τ can promote x no more, and so gives it open permission: we let $R_s(x) = R_{s-1}(x)\backslash\{\tau\}$. The parameters $k^\rho_s(x)$, $M^\rho_s(x)$ and $t^\rho_s(x)$ remain unchanged for all $\rho \in R_s(x)$. We choose a new value for $\texttt{top}(x)$: we let $\texttt{top}_s(x)$ be the longest $\rho \in R_s(x)$ for which $k^\rho_s(x)$ is maximal among the elements of $R_s(x)$.

C. Otherwise, we let τ promote x. We let $R_s(x) = R_{s-1}(x)$, and $k^\tau_s(x) = k^\tau_{s-1}(x) - 1$. We let $M^\tau_s(x) = B^\tau(k, \beta^\tau_s(k)\hat{}m^\tau_s(k))$ for $k = k^\tau_s(x)$. We let $t^\tau_s(x) = s$. We define $\psi^\tau_s(z) = s$ for all $z \in M^\tau_s(x)$. The parameters $k^\rho_s(x)$, $M^\rho_s(x)$ and $t^\rho_s(x)$ remain unchanged for other nodes $\rho \in R_s(x)$. We do, however, pick a new value for $\texttt{top}(x)$ as we did in the second case: we let $\texttt{top}_s(x)$ be the longest $\rho \in R_s(x)$ for which $k^\rho_s(x)$ is maximal among the elements of $R_s(x)$.

We then end the stage.

Suppose now that $\sigma \in \mathcal{P}$ is accessible at stage s. If there is some follower for σ which is still unrealised at stage s, then σ takes no action, and lets its only child be accessible (unless $|\sigma| = s$, in which case we end the stage). Otherwise, σ appoints a new follower x, of large value. We cancel all followers for all nodes weaker than σ. We then set up x's parameters as follows:

- We let $R_s(x)$ be the collection of all nodes $\tau \in \mathcal{N}$ such that $\tau\hat{}\infty \subseteq \sigma$. By the way we distributed the requirements on the tree, we see that $R_s(x)$ is nonempty.
- We let $\texttt{top}_s(x)$ be the longest node in $R_s(x)$.
- For all $\tau \in R_s(x)$, we let $k^\tau_s(x) = |\sigma|$. Note, of course, that as $\tau \subset \sigma$ we get $k^\tau_s(x) \geqslant |\tau|$.
- We let $t^\tau_s(x) = s$ for all $\tau \in R_s(x)$.
- For $\tau = \texttt{top}_s(x)$, we let $M^\tau_s(x) = \{z^\tau(\sigma)\}$, the singleton subset of $J^\tau(|\sigma|)$ which is reserved for σ.
- For $\tau \in R_s(x)\backslash\{\texttt{top}_s(x)\}$ we let $M^\tau_s(x) = B^\tau(|\sigma|, \beta^\tau_s(|\sigma|)\hat{}m^\tau_s(|\sigma|))$.

For all $\tau \in R_s(x)$, for all $z \in M^\tau_s(x)$, we set $\psi^\tau_s(z) = s$. We then end the stage.

3.3. **Justification.** Again, we fix a stage s^*, assume that $\otimes_s$ holds for all $s < s^*$, perform the construction up to stage s^*, and show that $\otimes_{s^*}$ holds as well. Much of the argument mimics the argument given in the previous section, and so we give the definitions and notation, and then only highlight the new ingredients. We start though by tracking the possible combinations for the function $\tau \mapsto k^\tau_s(x)$ on $R_s(x)$.

For $\sigma \in \mathcal{P}$ and $s < s^*$, we let F^σ_s be the collection of followers for σ which are alive at the end of stage s. We let $F_s = \bigcup_{\sigma\in\mathcal{P}} F^\sigma_s$.

We note that if $r < s$ and $x \in F_r \cap F_s$, then $R_s(x) \subseteq R_r(x)$. If $\tau \in R_s(x)$ and $k_s^\tau(x) = k_r^\tau(x)$, then $M_s^\tau(x) = M_r^\tau(x)$ and $t_s^\tau(x) = t_r^\tau(x)$. Hence for $k = k_s^\tau(x)$ we let $M^\tau(k,x) = M_s^\tau(x)$ and $t^\tau(k,x) = t_s^\tau(x)$. If $M^\tau(k,x) \neq \{z^\tau(\sigma(x))\}$, then we let $\alpha^\tau(k,x) = \alpha_s^\tau(x)$.

Lemma 3.1. *Let $x \in F_s$, and let $k = k_s^{\mathtt{top}_s(x)}(x)$. Let $\tau \in R_s(x)$.*

1. *If $\tau \subseteq \mathtt{top}_s(x)$, then $k_s^\tau(x) = k$.*
2. *If $\mathtt{top}_s(x) \subsetneq \tau$, then $k_s^\tau(x) = k - 1$.*

Also, if $\tau \in R_{s-1}(x) \backslash R_s(x)$, then τ is the longest string in $R_{s-1}(x)$.

Proof. The last part of the lemma follows from (1) and (2). To see this, suppose that $\tau \in R_{s-1}(x)\backslash R_s(x)$; so $\tau = \mathtt{top}_{s-1}(x)$ and x is promoted at stage s. We have $k = k_{s-1}^\tau(x) = |\tau|$. Suppose that $\rho \in R_{s-1}(x)$. If $\tau \subsetneq \rho$, then by the first part of the lemma, applied at stage $s-1$, we have $k_{s-1}^\rho(x) = k - 1$, which is smaller than $|\rho|$, which is impossible.

Now we prove (1) and (2), by induction on s.

If x is appointed at stage s, then for all $\tau \in R_s(x)$, $k_s^\tau(x) = k = |\sigma(x)|$, and $\mathtt{top}_s(x)$ is the longest element of $R_s(x)$.

Suppose that x is promoted at stage s. Let $\overline{\tau} = \mathtt{top}_{s-1}(x)$ and let $\overline{k} = k_{s-1}^{\overline{\tau}}(x)$. For all $\tau \in R_s(x)$ different from $\overline{\tau}$ we have $k_s^\tau(x) = k_{s-1}^\tau(x)$.

If $\overline{\tau} \notin R_s(x)$, then by the last part of the lemma (applied at stage $s-1$), $\overline{\tau}$ is the longest node in $R_{s-1}(x)$. Hence for all $\tau \in R_s(x)$ we have $k_s^\tau(x) = \overline{k}$, and $\mathtt{top}_s(x)$ is chosen to be the longest string in $R_s(x)$.

Suppose then that $\overline{\tau} \in R_s(x)$, so $R_s(x) = R_{s-1}(x)$. We set $k_s^{\overline{\tau}}(x) = \overline{k} - 1$. There are two cases.

- If $\overline{\tau}$ is the shortest string in $R_s(x)$, then for all $\tau \in R_s(x)$ we have $k_s^\tau(x) = \overline{k} - 1$; we then choose $\mathtt{top}_s(x)$ to be the longest node in $R_s(x)$.
- Otherwise, for $\tau \in R_s(x)$, if $\tau \supseteq \overline{\tau}$ then $k_s^\tau(x) = \overline{k} - 1$, and if $\tau \subsetneq \overline{\tau}$ then $k_s^\tau(x) = \overline{k}$. We choose $\mathtt{top}_s(x)$ to be the immediate predecessor of $\overline{\tau}$ in $R_s(x)$.

□

In particular, Lemma 3.1 shows that if x is promoted at stage s, then $\mathtt{top}_s(x) \neq \mathtt{top}_{s-1}(x)$.

Let $\tau \in \mathcal{N}$, $k \geqslant |\tau|$ and $s < \omega$.

- We let $K_s^\tau(k)$ be the collection of followers $x \in F_s$ such that $\tau = \mathtt{top}_s(x)$, $k = k_s^\tau(x)$, and $M^\tau(k,x) \neq \{z^\tau(\sigma(x))\}$.
- We let $L_s^\tau(k)$ be the collection of followers $x \in F_s$ such that $\tau = \mathtt{top}_s(x)$, $k = k_s^\tau(x)$, and $M^\tau(k,x) = \{z^\tau(\sigma(x))\}$.
- We let $G_s^\tau(k)$ be the collection of followers $x \in F_s$ such that $\tau \in R_s(x)$, $k = k_s^\tau(x)$, but $\tau \neq \mathtt{top}_s(x)$.

Again we use the notation $KG_s^\tau(k)$ to denote the union $K_s^\tau(k) \cup G_s^\tau(k)$, and the notation $\overline{K_s^\tau}(k) = K_{s-1}^\tau(k) \cap K_s^\tau(k)$ etc. Again, $KLG_s^\tau(k)$ is the collection of followers $x \in F_s$ such that $\tau \in R_s(x)$ and $k = k_s^\tau(x)$, $KL_s^\tau(k)$ is the collection of followers $x \in KLG_s^\tau(k)$ such that $\tau = \mathtt{top}_s(x)$, and $KG_s^\tau(k)$ is the collection of followers $x \in KLG_s^\tau(k)$ such that $\alpha^\tau(k,x)$ is defined.

The following lemma translates the construction into this terminology:

Lemma 3.2. *Let $x \in KLG_s^\tau(k)$. Let $t = t^\tau(k,x)$. The stage t is the least stage at which $x \in KLG^\tau(k)$. At stage t, x is placed into $LG^\tau(k)$.*

- *If x was appointed at stage t, then x is placed into $L_t^\tau(k)$ if $\tau = \texttt{top}_t(x)$, and into $G_t^\tau(k)$ if not.*
- *Otherwise, x is realised at stage t, and x is added to $G_t^\tau(k)$.*

Unless $x \in L_t^\tau(k)$, at stage t we let $M^\tau(k,x) = B^\tau(k, \beta_t^\tau(k)\hat{\ } m_t^\tau(k))$. The string $\beta_t^\tau(k)$ is defined as follows:

- *If $\overline{K_t^\tau}(k) = \varnothing$, then $\beta_t^\tau(k) = \langle\rangle$.*
- *If $\overline{K_t^\tau}(k) \neq \varnothing$, then $\beta_t^\tau(k) = \alpha^\tau(k, \max \overline{K_t^\tau}(k))$.*

The number $m_t^\tau(k)$ is the least $m < a(k)$ such that for all $y \in \overline{KG_t^\tau}(k)$, $\alpha^\tau(k,y) \neq \beta_t^\tau(k)\hat{\ } m_t^\tau(k)$.

The arguments of the justification for the minimal pair construction of the previous section now carry through, where $i < 2$ is replaced by $\tau \in \mathcal{N}$ and *stage* is replaced by *τ-stage*. This gives us analogues of all lemmas from Lemma 2.3 to Lemma 2.10, including the first part of $\otimes_{s*}$, that $|\beta_{s*}^\tau(k)| \leqslant k$.

Lemma 3.3. *For all $\sigma \in \Theta^\tau(k)$, $|F_s^\sigma \cap L_s^\tau(k)| \leqslant k+1$.*

Proof. Identical to the proof of Lemma 2.12. □

Let $b(k) = (k+1)(1+2^k)$. So $a(k) = 1 + 2^{k+2} \cdot b(k+1)$. Since $|\Theta^\tau(k)| \leqslant 2^k$, Lemma 3.3 and the analogue of Lemma 2.10 tell us that $|KL_s^\tau(k)| \leqslant b(k)$.

Lemma 3.4. *$|G_s^\tau(k)| < a(k)$.*

Proof. Let $x \in G_s^\tau(k)$; let $\rho = \texttt{top}_s(x)$. By Lemma 3.1, $x \in KL_s^\rho(k)$ or $x \in KL_s^\rho(k+1)$. This implies that $|\rho| \leqslant k+1$, so there are at most 2^{k+2} many possibilities for such strings ρ. We have seen that $|KL_s^\rho(k)|$ and $|KL_s^\rho(k+1)|$ are bounded by $b(k)$ and $b(k+1)$ respectively, and so both bounded by $b(k+1)$. Hence $|G_s^\tau(k)| \leqslant 2^{k+2} b(k+1) = a(k) - 1$. □

The argument of the previous section now gives a proof of $\otimes_{s*}$.

3.4. **Verification.**

Lemma 3.5. *Every follower receives attention only finitely many times.*

Proof. The same as the proof of Lemma 2.15, using $k_s(x) = \sum_{\tau \in R_s(x)} k_s^\tau(x)$. □

The analogue of Lemma 2.14 is the fact that the true path is infinite. Recall that a node ρ lies on the true path if ρ is accessible at infinitely many stages, but there are only finitely many stages at which some node that lies to the lexicographic left of ρ is accessible. The true path is a linearly ordered initial segment of the tree of strategies.

Lemma 3.6. *If $\tau \in \mathcal{N}$ lies on the true path, then one of τ's children lies on the true path as well.*

Proof. Suppose, for contradiction, that the lemma fails. This means that there is some stage s_0 such that for every $s \geqslant s_0$, if τ is accessible at stage s then s is a *τ-stage* and τ permits some follower at stage s.

Let $KLG_s^\tau = \bigcup_{k \geqslant |\tau|} KLG_s^\tau(k)$. Every $x \in KLG_s^\tau$ was appointed by the node $\sigma(x) \supseteq \tau\hat{}\infty$. Thus, there are only finitely many followers in $\bigcup_s KLG_s^\tau$. Eventually, none of them receive attention, contradiction. □

Again, for $X \in \{K, L, G\}$, let $X_\omega^\tau(k) = \lim_s X_s^\tau(k)$. For each $\tau \in \mathcal{N}$, $k \geqslant |\tau|$ and $X \in \{K, L, G\}$, we see that $X_\omega^\tau(k)$ is finite.

Lemma 3.7. *Suppose that $\sigma \in \mathcal{P}^e$ lies on the true path. Then the requirement P^e is met, and there is a stage after which no follower for $\mathcal{P}^e$ ever requires attention. The unique child of σ also lies on the true path.*

Proof. This is proved by induction on $|\sigma|$, using the argument of Lemma 2.16. □

As a corollary, we see that the true path is infinite.

Lemma 3.8. *Let $\tau \in \mathcal{N}^{e,c}$ be on the true path. Suppose that $T^{e,c}$ is a trace for ψ^e. Then $\tau\hat{}\infty$ lies on the true path.*

Proof. By Lemma 3.6, it suffices to show that there are infinitely many τ-*stages*. This follows from the fact that T^τ traces ψ^τ. □

Lemma 3.9. *E is not computable.*

Proof. We need to show that every requirement P^e is met. By Lemma 3.7, it suffices to show that for all e there is some node $\sigma \in \mathcal{P}^e$ on the true path. This follows from the fact that the true path is infinite, and from the way we distributed requirements to nodes, once we see that there is some node $\tau \in \mathcal{N}$ such that $\tau\hat{}\infty$ lies on the true path. This follows from Lemma 3.8, and the fact that SJT-hard c.e. sets exist. □

Reductions. We turn to show that E is computable from every SJT-hard c.e. set. Suppose that W^e is SJT-hard; so there is some $c < \omega$ such that $T^{e,c}$ traces ψ^e. As the true path is infinite, find some $\tau \in \mathcal{N}^{e,c}$ on the true path. By Lemma 3.8, $\tau\hat{}\infty$ lies on the true path; there are infinitely many τ-*stages*.

Lemma 3.10. *Let $x \in F_s$ and suppose that $\tau \in R_s(x)$. If $x \in E$, then there is some stage $r \geqslant s$ such that $\tau = \mathtt{top}_r(x)$.*

Proof. Let $v > s$ be the stage at which x is enumerated into E. We have $R_{v-1}(x) = \{\mathtt{top}_{v-1}(x)\}$. If $\tau = \mathtt{top}_{v-1}(x)$ we are done. Otherwise, since $\tau \notin R_{v-1}(x)$, let $r \geqslant s$ be the last stage at which $\tau \in R_r(x)$. Then $\tau = \mathtt{top}_r(x)$. □

Lemma 3.11. *Let s be a τ*-stage. *Let $x \in F_s$ such that $\tau \in R_s(x)$ and suppose that $v_s^\tau(x)\downarrow$. Suppose that $W^e \upharpoonright v_s^\tau(x) = W_s^e \upharpoonright v_s^\tau(x)$. Then $x \notin E$.*

Proof. Identical to the proof of Lemma 2.17, using Lemma 3.10. □

Let H be the collection of followers which are never cancelled nor enumerated into E. We use notation, such as $R_\omega(x)$, similar to the notation we used before.

Lemma 3.12. *Let $x \in H$ and let $\tau \in R_\omega(x)$. There is some τ*-stage *s such that $x \in F_s$, $v_s^\tau(x)\downarrow$ and $W^e \upharpoonright v_s^\tau(x) = W_s^e \upharpoonright v_s^\tau(x)$.*

Proof. Identical to the proof of Lemma 2.18, using the assumption that T^τ traces ψ^τ. □

Lemma 3.13. *$E \leqslant_{\mathrm{T}} W^e$.*

Proof. Similar to the proofs of Lemmas 2.19 and 2.21.

Let $x \in F_x$. To find, with oracle W^e, whether $x \in E$ or not, wait for a τ-*stage* $s > x$ at which one of the following holds:

- x has been cancelled by stage s;
- $x \in E_s$; or
- $\tau \in R_s(x)$, $v_s^\tau(x)\downarrow$ and $W^e \upharpoonright v_s^\tau(x) = W_s^e \upharpoonright v_s^\tau(x)$.

If such a stage s is found, then by Lemma 2.17, W^e can decide at stage s if $x \in E$ or not.

We claim that such a stage s can be found for all but finitely many followers x. First, note that there are only finitely many followers $x \in \bigcup_s F_s$ such that $\sigma(x)$ is stronger than $\tau\hat{\ }\infty$: those σ that lie to the left of τ are visited only finitely many times, and those that are extended by τ lie on the true path, and so appoint only finitely many followers by Lemma 3.7.

Hence, we let $x \in \bigcup_s F_s$ and assume that $\sigma(x)$ is not stronger than $\tau\hat{\ }\infty$. If $\sigma(x)$ lies to the right of $\tau\hat{\ }\infty$, then any $\tau\hat{\ }\infty$ stage $s > x$ satisfies the condition above: either x is cancelled by stage s, or $x \in E_s$. Suppose, then, that $\sigma(x) \supseteq \tau\hat{\ }\infty$. Thus, $\tau \in R_t(x)$ where t is the stage at which x is appointed.

If $x \notin H$ then a stage as above is definitely found. Suppose that $x \in H$. If τ is never removed from $R(x)$, then Lemma 3.12 ensures that a stage s as above is found. Otherwise, let $\rho = \mathtt{top}_\omega(x)$. Let $r + 1$ be the stage at which τ is removed from $R(x)$; so $k_r^\tau(x) = |\tau|$. Lemma 3.1 shows that $\rho \subset \tau$, and that $k_r^\rho(x) = |\tau|$. So $k_\omega^\rho(x) \leqslant |\tau|$. Thus

$$x \in \bigcup_{\rho \subset \tau, \rho \in \mathcal{N}} \bigcup_{k \leqslant |\tau|} KL_\omega^\rho(k).$$

However, this set is the finite union of finite sets, and so is finite. ∎

References

[1] Douglas Cenzer, Geoffrey Laforte, and Guohua Wu. Pseudojumps and Π_1^0 classes. *J. Logic Comput.*, 19(1):77–87, 2009.

[2] Peter Cholak, Rodney G. Downey, and Noam Greenberg. Strong jump-traceabilty I: The computably enumerable case. *Adv. Math.*, 217(5):2045–2074, 2008.

[3] Richard J. Coles, Rodney G. Downey, Carl G. Jockusch, Jr., and Geoffrey Laforte. Completing pseudojump operators. *Ann. Pure Appl. Logic*, 136(3):297–333, 2005.

[4] Karel de Leeuw, Edward F. Moore, Claude E. Shannon, and Norman Z. Shapiro. Computability by probabilistic machines. In *Automata studies*, Annals of mathematical studies, no. 34, pages 183–212. Princeton University Press, Princeton, N. J., 1956.

[5] Natasha L. Dobrinen and Stephen G. Simpson. Almost everywhere domination. *J. Symbolic Logic*, 69(3):914–922, 2004.

[6] Rodney G. Downey and Noam Greenberg. Strong jump-traceability II: K-triviality. To appear in the Israel Journal of Mathematics.

[7] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic randomness and complexity*. Theory and Applications of Computability. Springer, New York, 2010.

[8] Rodney G. Downey, Denis R. Hirschfeldt, André Nies, and Frank Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore Univ. Press.

[9] Rodney G. Downey, Denis R. Hirschfeldt, André Nies, and Sebastiaan A. Terwijn. Calibrating randomness. *Bull. Symbolic Logic*, 12(3):411–491, 2006.

[10] Santiago Figueira, André Nies, and Frank Stephan. Lowness properties and approximations of the jump. *Ann. Pure Appl. Logic*, 152(1-3):51–66, 2008.

[11] Noam Greenberg, Denis R. Hirschfeldt and André Nies. Characterising the strongly jump-traceable sets via randomness. Submitted.

[12] Noam Greenberg and André Nies. Benign cost functions and lowness properties. *J. Symbolic Logic*, 76(1):289–312, 2011.
[13] Noam Greenberg and Daniel Turetsky. Strong jump-traceability and Demuth randomness. Submitted.
[14] Carl G. Jockusch, Jr. and Richard A. Shore. Pseudojump operators. I. The r.e. case. *Trans. Amer. Math. Soc.*, 275(2):599–609, 1983.
[15] Carl G. Jockusch, Jr. and Richard A. Shore. Pseudojump operators. II. Transfinite iterations, hierarchies and minimal covers. *J. Symbolic Logic*, 49(4):1205–1236, 1984.
[16] Bjørn Kjos-Hanssen, Joseph S. Miller and D. Reed Solomon. Lowness notions, measure and domination. To be published by the London Mathematical Society.
[17] Antonín Kučera and André Nies. Demuth randomness and computational complexity. *Ann. Pure Appl. Logic*, 162(7):504–513, 2011.
[18] Joseph S. Miller and André Nies. Randomness and computability: open questions. *Bull. Symbolic Logic*, 12(3):390–410, 2006.
[19] Webb Miller and Donald A. Martin. The degrees of hyperimmune sets. *Z. Math. Logik Grundlagen Math.*, 14:159–166, 1968.
[20] Keng Meng Ng. On strongly jump traceable reals. *Ann. Pure Appl. Logic*, 154(1):51–69, 2008.
[21] Keng Meng Ng. On very high degrees. *J. Symbolic Logic*, 73(1):309–342, 2008.
[22] André Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.
[23] André Nies. Eliminating concepts. In *Computational prospects of infinity. Part II. Presented talks*, volume 15 of *Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap.*, pages 225–247. World Sci. Publ., Hackensack, NJ, 2008.
[24] André Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009.
[25] André Nies. Calculus of cost functions. In preparation.
[26] Jean Raisonnier. A mathematical proof of S. Shelah's theorem on the measure problem and related results. *Israel J. Math.*, 48(1):48–56, 1984.
[27] Robert M. Solovay. Draft of paper (or series of papers) related to Chaitin's work. IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 215 pages, 1975.
[28] Clifford Spector. Measure-theoretic construction of incomparable hyperdegrees. *J. Symb. Logic*, 23:280–288, 1958.
[29] Sebastiaan A. Terwijn. *Computability and Measure*. PhD thesis, University of Amsterdam, 1998.
[30] Sebastiaan A. Terwijn and Domenico Zambella. Computational randomness and lowness. *J. Symbolic Logic*, 66(3):1199–1205, 2001.

School of Mathematics, Statistics and Operations Research, Victoria University of Wellington, Wellington, New Zealand
E-mail address: downey@msor.vuw.ac.nz
URL: http://homepages.mcs.vuw.ac.nz/~downey/

School of Mathematics, Statistics and Operations Research, Victoria University of Wellington, Wellington, New Zealand
E-mail address: greenberg@msor.vuw.ac.nz
URL: http://homepages.mcs.vuw.ac.nz/~greenberg/