August 24, 2011

PSEUDO-JUMP INVERSION AND SJT-HARD SETS

ROD DOWNEY AND NOAM GREENBERG

Abstract. There are noncomputable c.e. sets, computable from every SJT-hard c.e. set. 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). In the same spirit, Terwijn and Zambella used a uniform version of hyperimmunity 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, 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. Thus, rather than using the exact size of each element of the trace, we use effective bounds on this size to indicate how strong a trace is: the fewer options for the value of a function, the closer we are to knowing what that value is. 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*, the uniform version of hyperimmunity used by Terwijn and Zambella, is defined by requiring that traces for functions in a hyperimmune 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 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 (Figuiera, 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 sets, 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 $A \in 2^\omega$, $J(A) >_{\mathrm{T}} A$.

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

The most natural pseudo-jump operator is 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 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*. 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 B-partial computable function ψ has an A-c.e. trace 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 A-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 non-computable 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 any order function such that $\sum 2^{-h(n)}$ is computable and finite; the proof of Theorem 1.8 gives an order function g whose growth rate is roughly $\log\log n$. The gap does not seem too large, and at least one of the authors believes that it can be bridged.

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 next to nothing. Further research is definitely needed. This 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. As a warm-up, we prove a weaker result first: that there is no minimal pair of c.e. SJT-hard degrees.

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 $A^1 \restriction u^1$ more or less at this stage, 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. Furthermore, whilst u^0 and u^1 are chosen to be large when x is appointed, a long time passes between that stage and the stage at which x is realised; of course, before x is realised, we do not want to enumerate it into E. Hence, when we want to enumerate x into E, the uses u^0 and u^1 would be relatively small, and so voluntary A^0 and A^1 changes below their uses are unlikely.

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 $T^i = \left\langle T^i(z) \right\rangle_{z<\omega}$ 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, $|T^i(z)| \leqslant h(z)$. Now a Σ^0_2 function ψ is the partial limit of a sequence $\langle \psi_s \rangle_{s<\omega}$ of uniformly computable functions; $\psi(z) = d$ if and only if for almost all s, $\psi_s(z) = d$. The trace T^i is uniformly c.e. in A^i, that is, every $d \in T^i(z)$ is enumerated into $T^i(z)$ with some A^i-use, which we denote by $u^i(z, d)$. At stage s, we let $T^i_s(z)$ be the collection of numbers enumerated into $T^i(z)$ by stage s with the oracle A^i_s, the collection of numbers already in A^i at stage s. An important point is that if $\psi_s(z) = d$ at some stage s, then either $d \in T^i(z)$, or there is some stage $t > s$ such that $\psi_t(z) \neq d$. Thus, if we are defining ψ, then when we let $\psi_s(z) = d$, we can commit to never changing the value $\psi_t(z)$ away from d unless we see that $d \in T^i_t(z)$. This implies that for all s and z, there is some $t > s$ such that $\psi_s(z) \in T^i_t(z)$. If $d = \psi_s(z)$ is an element of $T^i_t(z)$, with A^i-use $u = u^i_t(z, d)$, then at a stage $r > t$, either d is still an element of $T^i_r(z)$, or $A^i_t \restriction u \neq A^i_r \restriction u$. In other words, extracting d from $T^i(z)$ requires an A^i-change below u. In particular, this means that if at stage t (or at a later stage) we change $\psi_t(z)$ to have value different from d, then either this new value needs to be enumerated into $T^i(z)$ in addition to the old value d, or the old value needs to be extracted from $T^i(z)$ by an A^i-change below u. If we repeat this process more than $h(z)$ many times, then as $T^i(z)$ cannot contain $h(z) + 1$ many distinct possible values for $\psi(z)$, at some cycle we will have forced a change in A^i. The smaller the value $h(z)$ is, the fewer cycles we need to go through before we obtain the desired A^i-change.

The construction below is a balancing act. We will show that if h grows sufficiently slowly, then we will be able to force sufficiently many simultaneous A^0- and A^1-changes, to get enough simultaneous permissions so that for every requirement P^e there is *some* follower x for P^e which is permitted to enter E (or is never realised). So as described above, we define a Σ^0_2 partial function ψ, by letting ψ be the partial limit of a uniformly computable sequence $\langle \psi_s \rangle$ of functions. We also define an order function h. As we argue below, the recursion theorem will give us, for both $i = 0, 1$, A^i-c.e. traces $T^i = \left\langle T^i(z) \right\rangle_{z<\omega}$ for ψ which are both bounded by h. As described above, we let $T^i_s(z)$ be the stage s-approximation for $T^i(z)$, and we assume that for all s and z there is some $t > s$ such that $\psi_s(z) \in T^i_t(z)$.

The general process for obtaining an A^i-permission for some follower x for P^e follows the procedure, described above, of manufacturing an A^i-change. We associate some input z with the requirement P^e. At some stage s_0, we appoint a first follower x_0 for P^e, and define $\psi_{s_0}(z) = s_0$. We then wait for a stage $t_0 > s_0$ at which we see that s_0 is enumerated into $T^i(z)$. This enumeration yields an A^i-use $u^i_{t_0}(z, s_0)$; we determine that the A^i-use $v^i(x_0)$ which is used for the reduction of the question "$x_0 \in E$?" to A^i equals the use $u^i_{t_0}(z, s_0)$. Now suppose that at some stage $s_1 > t_0$, the follower x_0 is realised; so for A^i-permission for x_0, we need to see a future change in $A^i \upharpoonright v^i(x_0)$. The definition of $v^i(x_0)$ means that if A^i does not give permission to x_0 by a stage r, then $s_0 \in T^i_r(z)$. We then appoint a new follower x_1, and redefine $\psi_{s_1}(z)$ to equal s_1. We wait for a stage $t_1 > s_1$ at which we see that s_1 is enumerated into $T^i(z)$. Now there are two possibilities: if $A^i_{t_1} \upharpoonright v^i(x_0) \neq A^i_{t_0} \upharpoonright v^i(x_0)$, then x_0 receives A^i-permission as required. Otherwise, we define the use $v^i(x_1) = u^i_{t_1}(z, s_1)$ for reducing the question "$x_1 \in E$?" to A^i; we wait for another stage s_2 at which x_1 is realised, and repeat the process: appoint a new follower x_2, change $\psi_{s_2}(z)$ to equal s_2, find a stage $t_2 > s_2$ at which s_2 is enumerated into $T^i(z)$. At stage t_2, if neither x_0 nor x_1 are permitted by A^i, then we appoint a new follower and repeat the cycle. As we mentioned above, this process cannot repeat more than $h(z) + 1$ many times, and so eventually, either we appoint a follower which never gets realised, or some realised follower is permitted by A^i.

The reader may ask: after all, we define h. Why do we not just define $h(z) = 1$? In this case, we do not need to appoint more than one follower, as the first follower appointed is guaranteed to receive permission. There are two reasons. First, while we define h, the recursion theorem levies an "overhead", which means that it gives us a constant $c \geqslant 1$ such that T^i is only bounded by $\max\{c, h\}$ and not by h. The second reason is more important. While we define h, we need to ensure that h is unbounded. That means that for any value k, we may only define $h(z) = k$ for finitely many inputs z. Fixing z, if x is a follower for P^e which is never realised, we will not need permission from A^i for x, but we also need to ensure that the use of reducing $x \in E$ to A^i is bounded. If we let other followers, for other requirements, access z by changing $\psi(z)$ and thus the A^i-use of enumerating $\psi(z)$ into $T^i(z)$, then the use of reducing $x \in E$ to A^i, which has to be tied to this use (to enable permission if x does get realised), will increase each time $\psi(z)$ is changed. This should not happen infinitely often. And so, for yet unrealised followers, different requirements P^e need to access different inputs z, and so the values $h(z)$ for the inputs z cannot be bounded.

The discussion so far shows how to obtain permission from a single set A^0 or A^1, and so would suffice if we wanted to show that both sets are not computable. However, to enumerate a follower x into E, we need permission from both sets A^0 and A^1. If we did have "1-boxes", namely access to inputs z such that $|T^i(z)|$ is known to be bounded by 1, then we could run the process above on both sides: appoint a follower x at a stage s; define $\psi_s(z) = s$; for both $i < 2$, tie the A^i-use of x to the A^i-use $v^i(x) = u^i_t(z, s)$ of enumerating s into $T^i(z)$ at some stage $t > s$; if x is realised at some $s' > t$, redefine $\psi(z) = s'$, get permission for x from both A^0 and A^1, and enumerate x into E. The point here is that if we have 1-boxes then the *only* way a new value can enter the box is that the relevant A^i changes on the use. However, we may not have 1-boxes. If $h(z) > 1$, and we attempt to run the process on both sides, we may be faced with the following situation which is caused by certain *timing* considerations. At stage s_0, we appoint the follower x_0 and define $\psi_{s_0}(z) = s_0$; at stage $t_0 > s_0$, we associate with x_0 the A^i-uses $v^i(x_0) = u^i_{t_0}(z, s_0)$. At stage $s_1 > t_0$, x_0 is realised; we appoint a new follower x_1, set $\psi_{s_1}(z) = s_1$, and wait for s_1 to be enumerated into both $T^0(z)$ and $T^1(z)$ at a stage $t_1 > s_1$. At stage t_1, we see that A^0 permits x_0, but A^1 does not. We are not allowed to enumerate x into E at stage t_1. But we are also not allowed to *keep the A^0-permission for x open*: at stage t_1, we must associate x_0 with a new A^0-use, or we will not be able to compute the answer to the question "$x \in E$?" from A^0 alone. We can do this; we redefine $u^0(x_0) = u^0(x_1) = u^0_{t_1}(z, s_1)$. Say that at a later stage s_2, x_1 is realised as well. Suppose, for simplicity, that $h(z) = 2$. Thus if we change $\psi(z)$ at stage s_2, then we are guaranteed that A^1 permits either x_0 or x_1 at some $t_2 > s_2$. But the A^0-permission for x_0 at stage t_1 means that s_0 is extracted from $T^0(z)$, and so $T^0_{t_1}(z)$ might contain but one element, s_1. In this case, A^0 may refuse to permit either x_0 or x_1 at stage t_2. The permission that A^1 gives at stage t_2, though, gives A^1 the opportunity to empty $T^1(z)$ and leave it with one element, s_2. This strategy allows A^0 and A^1 together to keep see-sawing, always giving permission on one side, and denying it on the other. Hence whilst the opponent gives permissions on both sides, because they are out of phase, we cannot use them for enumeration of x_0, or indeed any other follower, into E.

In fact, if the growth rate of the order h is 2^n, then we know that it is possible to construct a minimal pair of c.e. sets B^1, B^2, which are superhigh with truth table bound 2^n. (See for example Ng's [21]). Thus somehow we will need to have a more slowly growing h, and this is "dually" similar to the problems in showing that the strongly jump-traceable c.e. sets form an ideal, as shown in Cholak, Downey and Greenberg [2].

We need to break the symmetry between A^0 and A^1. Using a slowly growing order function h, this is done by an inductive process known as "box promotion". This promotion has already been exploited. If some realised follower x is tied to an A^i-box $\{z\}$ via some value $s \in T^i(z)$ (and use $v^i(x) = u^i_t(z, s)$), we change $\psi(z)$ away from s, and $A^i \upharpoonright u$ does not change, then the $h(z)$-box $\{z\}$ has been *promoted* to being an $h(z) - 1$-box. Other followers, weaker than x, can use the box, imagining that it only has $h(z) - 1$ many slots left to fill. This promotion can be reversed only if A^i relents and permits x.

We sketch how this works. Suppose, as first approximation, that the constant c supplied by the recursion theorem equals 1, that is, we have access to 1-boxes. Since we only have finitely many 1-boxes, we consider a requirement P^e which initially only has access to 2-boxes. Suppose that x is a realised follower for P^e; at some stage s, the A^i-uses are $v^i(x) = u^i_t(z_i, t)$, where $\{z_0\}$ and $\{z_1\}$ are 2-boxes. We ask for A_1-permission for x by changing $\psi(z_1)$ away from t. While permission is denied, $\{z_1\}$ is effectively a 1-box, and other followers may use it to define uses and obtain permission from A_1. Since we do not want to drive the use $v^0(x)$ to infinity, other followers, for the time being, are not allowed to change $\psi(z_0)$. Once A^1-permission is received, we move the A^1, x-pointer from z_1 to a 1-box $\{w\}$, and obtain a new version of $v^1(x)$, based on the use of enumerating a new value $\psi(w)$ into $T^1(w)$. We then ask for A^0-permission for x, making the 2-box $\{z_0\}$ "active": while A^0-permission for x is denied, $\{z_0\}$ is effectively a 2-box for other followers, who may now change $\psi(z_0)$ with impunity, defining uses based on z_0 and obtaining permissions from A^0. Once A^0-permission for x is obtained, we can change $\psi(w)$ and get a guaranteed A^1-permission for x as well, and enumerate x into E. See Figures 1 and 2.

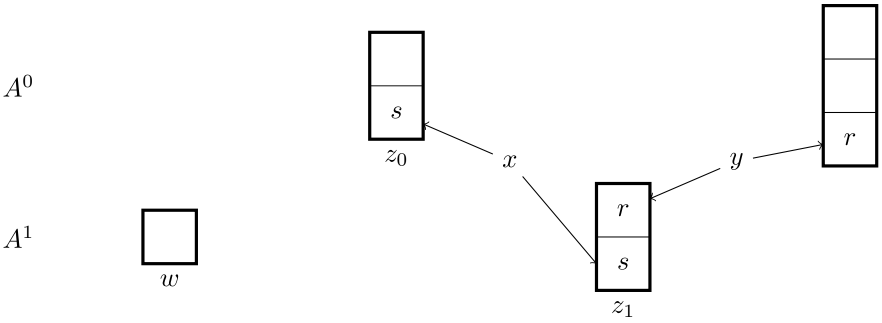

FIGURE 1. The follower x is waiting for an A^1-permission. Meanwhile, follower y can treat $\{z_1\}$ as a 1-box.

Now, an examination of what happens further to the right explains why we need to use *metaboxes.* A metabox is an aggregate of boxes of some fixed size. It is used to amplify the promotion given by a refusal of some set to give permission. Suppose again, for example, that both A^0- and A^1-uses of a follower x are associated with 2-boxes $\{z_0\}$ and $\{z_1\}$ as above. While waiting for A^1-permission, other followers treat $\{z_1\}$ as a 1-box. Such followers point to $\{z_1\}$ and to an $A^0 - 3$-box $\{z_2\}$. We need at least two such followers, y_0 and y_1, to turn $\{z_2\}$ into a 1-box. However, while waiting for permission from A^0, the A^1-use of y_0 and y_1, which is tied to $\{z_1\}$, should not change. This means that y_0 and y_1 should not be sharing z_1; they should have different versions of z_1, each for its own use. [The reader may ask: if there are only two followers, y_0 and y_1, then not much harm will come from changing, say, y_0's use when y_1 acts, as y_1 will act only once? But it is possible that y_0 is fixed, but infinitely many followers playing the role of y_1 come and go, each acting once. This is why y_0 needs a box of its own on the A^1 side.] The solution is to tie x's use to to a single input z_1, but to a metabox M of inputs. When the use is set up, at stage s say, we define $\psi(z) = s$ for all $z \in M$. We wait for s to show up in $T^1(z)$

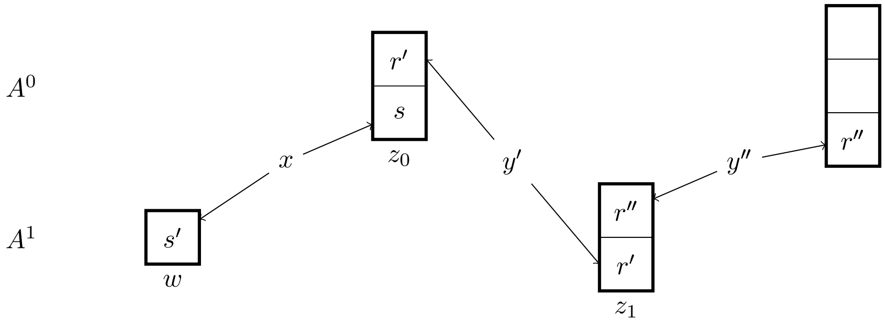

Figure 2. Follower x received permission from A^1. Its A^1-pointer moved to w. It is now waiting for A^0-permission. Meanwhile, follower y' is treating $\{z_0\}$ as a 1-box, and follower y'' is treating $\{z_1\}$ as a 1-box.

for all $z \in M$. We then let the use $v^1(x)$ be the maximum of the uses $u^1(z,s)$ of enumerating s into $T^1(z)$ for $z \in M$. While A^1 does not permit x, every $z \in M$ can be treated as a 1-box. This means that we can then split M into two, one part tied to y_0 and another to y_1. See Figure 3.

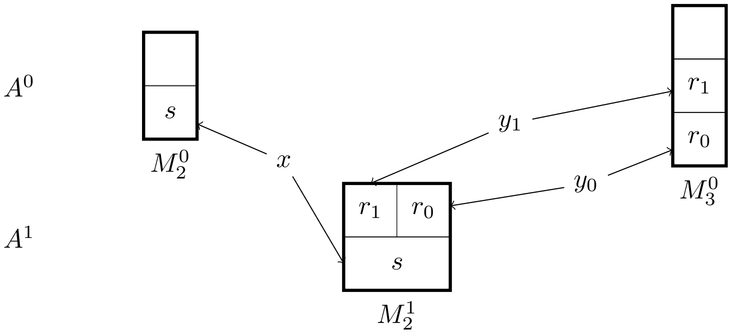

Figure 3. The metabox M^1_2 is promoted while x is waiting for A^1-permission. It is split between y_0 and y_1; while both y_0 and y_1 wait for A^0-permission, other followers can treat M^0_3 as a 1-box and get immediate permission when required.

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 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

[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, we believe that this would lead to a proof that there is no minimal pair of LR-hard c.e. sets.

stronger than y_1. If y_0 gets permission from A^0, and its A^0 pointer moves from M^0_3 to M^0_2, then y_1 gets cancelled, the part of M^1_2 which was pointed to by y_0 gets promoted, and it will be split up between future followers pointing at M^1_2 and M^0_3. The part of M^1_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^1_2 to which it is pointed 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.

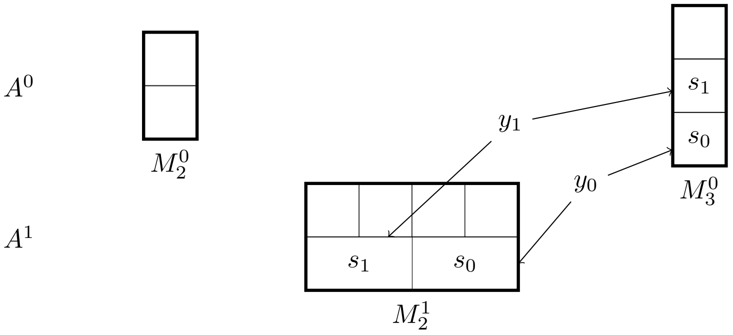

FIGURE 4. Both y_0 and y_1 seek permission from A^0. We do not know which, if either, will be permitted.

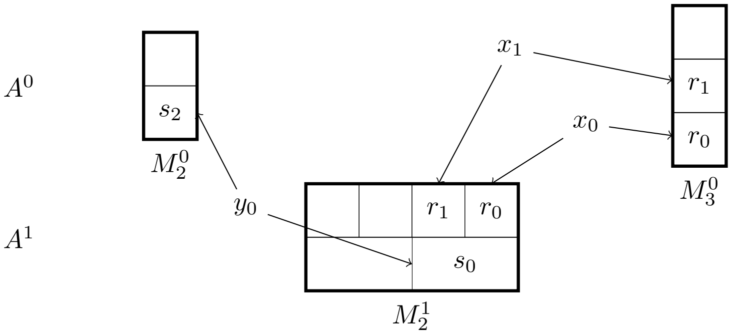

FIGURE 5. The follower y_0 was permitted. Followers x_0 and x_1 use y_0's boxes. y_1 is cancelled.

We see the general structure of the metaboxes M^i_k for $i < 2$ and $k < \omega$. Consider for example $i = 1$. A follower x, pointing at M^1_k, also points at either M^0_k or M^0_{k+1}. In the first case, it is asking for permission from A^1; in the second, from A^0. For such a follower x, let $M^1(x)$ be the sub-box of M^1_k which is used by x. If x and y point to M^1_k and M^0_{k+1} (we write $0 = \mathtt{top}(x) = \mathtt{top}(y)$ to indicate that x and y are waiting for permission from A^0), then we require that $M^1(x)$ and $M^1(y)$ are disjoint, so that the A^1-use of x, in case it never moves again, does not go to infinity. If x and y point to M^1_k and M^0_k (i.e. $1 = \mathtt{top}(x) = \mathtt{top}(y)$) and x is stronger than

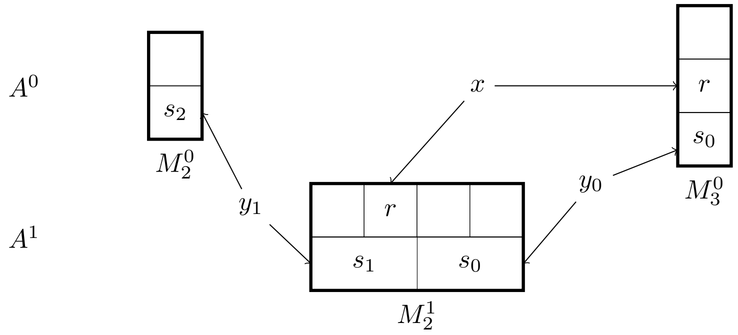

FIGURE 6. In an alternate reality, y_1 was permitted. A new follower x uses y_1's boxes.

y then we require that $M^1(y) \subseteq M^1(x)$, and that $t^1(x) < t^1(y)$. Here $t^1(x)$ is the stage at which x was last moved, which is the same as the value of $\psi(z)$ which is enumerated into $T^1(z)$ for $z \in M^1_k(x)$, the uses $u^1(z, t^1(x))$ of said enumerations determining the use connecting $x \in E$ to A^1. The conditions $M^1(y) \subseteq M^1(x)$ and $t^1(x) < t^1(y)$, together with the bound k on the size of $T^1(z)$ for $z \in M^1_k$, ensure that there are at most k followers x with $\mathtt{top}(x) = 1$ pointing at M^1_k. This allows us to calculate how large M^0_k needs to be (as we mentioned, about k^k), and so define the order function h ahead of time. The metabox M^1_k is structured as a tree of sub-boxes, ordered by reverse inclusion. A stem consists of the boxes $M^1(x)$ for followers x with $\mathtt{top}(x) = 1$; each element $M^1(x)$ of the stem is divided into disjoint sub-boxes $M^1(y)$, where for all but at most one y, we have $\mathtt{top}(y) = 0$.

For $i = 0$ instead of 1, the situation is similar; if x points at M^0_k (we write $k = k^0(x)$), then $\mathtt{top}(x) = 1$ if it also points at M^1_k, and $\mathtt{top}(x) = 0$ if it also points at M^1_{k-1}. A typical walk by a follower x starts with M^1_k and M^0_k for some k (so $\mathtt{top}(x)$ begins with 1); an A^1-permission moves x to point at M^0_k and M^1_{k-1} (moving $\mathtt{top}(x)$ to 0); then an A^0-permission moves x to point at M^1_{k-1} and M^0_{k-1} (moving $\mathtt{top}(x)$ back to 1), and so on, until at some point, one of the boxes it points to is a (possibly promoted) 1-box, which allows dual permission and enumeration into E.

This structure explains why we need to order the followers by priority, and cancel weak followers whenever a stronger follower moves. If x is stronger than y, then for both $i < 2$, the use $v^i(x)$ for reducing $x \in E$ to A_i is smaller than the corresponding use $v^i(y)$. If x receives permission from say A^0 at some stage, then the A^0-change below $v^0(y)$ invalidates any promotion of an A^0-box credited to y. On the other hand, on the A^1-side, the boxes $M^1(x)$ and $M^1(y)$ are disjoint; if y were not cancelled, we would have to move y to also require permission from A^1, violating the requirement that $M^1(x)$ and $M^1(y)$ are comparable in this case. And again we emphasise, that without this comparability, the whole mechanism, of using the bound on the size of the traces to bound the number of boxes that we need, is thrown out of whack.

To complete the discussion, there are three further issues we need to address. The first is simple: we notice that followers for different requirements need to all work in concert in the grand scheme of movement and promotion. That is, we cannot, for different requirements P^e, set up separate boxes, only devoted to followers of P^e. Again, this is because for each k, we can overall have only finitely many k-boxes, as h has to be unbounded. The *cursus honorum* of followers for P^e may start with e-boxes, say, but may require advancement all the way down to 1-boxes. The effort is combined; followers for a requirement P^e will make use of promotions credited to followers of stronger requirements. The important thing is that even though some followers get stuck, overall there are no losses but only collective gains.

The discussion above only referred to realised followers. Before a follower x is realised, it needs to point at boxes M^i_k for both $i < 2$, but permissions from A^1 where say $1 = \texttt{top}(x)$ are not useful, as we do not want to enumerate x into E before it is realised. So x cannot play a full part in the global promotion process. This is why we need to allocate to it a *private* A^1-box, disjoint from the general structure of sub-boxes of M^1_k. On the other hand, when x is realised and receives its first A^1-permission, it needs to join the global effort, while of course not moving its A^0-pointer. This is why the A^0-boxes associated with an unrealised follower are not private, but part of the general structure of boxes.

Finally, the impatient reader has likely already been wondering for a while, what if we do not have any 1-boxes? That is, what happens if the constant c supplied by the recursion theorem is greater than 1? This situation seems to invalidate the entire plan, since it loses its basis of 1-boxes: if we have none, then there is no mechanism for eventually enumerating any followers into E.

The solution is to introduce nonuniformity to the process of reducing E to A^1. Suppose that a follower x has been promoted as much as it could: it points to the smallest A^1-box M^1_c, and to the smallest A^0-box M^0_c, and $\texttt{top}(x) = 1$. It then receives A^1-permission, but we cannot guarantee A^0-permission, since $c > 1$. We then, contrary to everything we said so far, leave the A^1-permission open: we do not appoint a new A^1-use, and leave x to point only at M^0_c, and not at any A^1-box. The situation of x in the scheme of sub-boxes of M^0_c is as if it were pointing at M^1_{c-1}, if that metabox existed. So the boxes $M^0(x)$ of such followers are nested, and so there are at most c of them. If x is then permitted by A^0, we enumerate it into E; otherwise we do not. We then see that the promotion structure implies that there can be at most c followers for which the A^1-permission is left open but are not later cancelled or enumerated into E. Thus A^1, upon leaving a permission open for x, searches for a later stage at which x is cancelled or enumerated into E, and thus can find out if x is in E or not. This search will halt for all but finitely many, indeed at most c many, followers.

We note that the nonuniformity we introduced is limited to the reduction of E to A^1. Namely, an index for E is obtained effectively from indices for A^0 and A^1 (and for A^i-traces for a universal Σ^0_2 partial function), and also, an index for a reduction of E to A^0 is effectively obtained. 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 $\psi^i(z)\uparrow$.

As argued, for example, in [2], the recursion theorem provides us with a constant $c \geqslant 1$, and for both $i < 2$, an A_i-c.e. traces T^i which traces ψ^i. These traces are both bounded by an order function h, which we define; we may use the constant c in the definition, but we need to ensure that the definition of h is uniform in c, and that $h(0) \geqslant c$.

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, 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.

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.

With any follower x, "alive" at the end of a stage s (that means that x was appointed at some stage $s' \leqslant s$ and was not cancelled at any stage $s'' \in (s', s]$), we associate some auxiliary objects which aid with the reduction of the question "$x \in E$?" to A^0 and A^1.

- We attach a number $k^0_s(x) \in [c, e]$, and possibly also a number $k^1_s(x) \in [c, e]$. These are the *levels* at which the x-pointer points.
- We define a number $\mathtt{top}_s(x) \in \{0, 1\}$. This is the side from which x next requires permission. If $\mathtt{top}_s(x) = 1$ then $k^1_s(x)$ is defined.
- For $i < 2$ such that $k^i_s(x)$ is defined, we define a metabox $M^i_s(x) \subset I\left(k^i_s(x)\right)$.
- For such i, we also define a value $t^i_s(x)$. This is the value appearing in the boxes which we tie to the use of reducing $x \in E$ to A^i.

Let s be a *stage* at the end of which a follower x is alive. Let $i < 2$ such that $k^i_s(x)$ is defined. We say that an i-computation is defined for x at stage s if for all $z \in M^i_s(x)$ we have $t^i_s(x) \in T^i_s(z)$. In this case, we let

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

We denote the fact that an i-computation for x is defined at stage s by writing $v^i_s(x)\downarrow$. Otherwise, we write $v^i_s(x)\uparrow$.

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.

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. Let $a(k) = 2k+2$. We define a tree of sub-boxes of $I(k)$, indexed by strings in ${}^{\leqslant k+1}a(k)$. Let $J(k) = \{\min I(k)\}$; let $B(k, \langle\rangle) = I(k)\backslash J(k)$, where $\langle\rangle$ is the empty string. Note that $|B(k, \langle\rangle)| = a(k)^{k+1}$. Given $B(k, \alpha) \subset I(k)$ for $\alpha \in {}^{\leqslant k}a(k)$, which by induction has size $a(k)^{k+1-|\alpha|}$, we let

$$\{B(k, \alpha\hat{\ }m) \,:\, m < a(k)\}$$

be a partition of $B(k, \alpha)$ into $a(k)$ many subsets of equal size $a(k)^{k-|\alpha|}$. Thus we get a tree of boxes: for $\alpha, \beta \in {}^{\leqslant k+1}a(k)$, if $\alpha \subseteq \beta$ then $B(k, \alpha) \supseteq B(k, \beta)$, and if

$\alpha \perp \beta$ then $B(k,\alpha) \cap B(k,\beta) = \emptyset$. Also, for all $\alpha \in {}^{\leqslant k+1}a(k)$, $B(k,\alpha)$ is disjoint from $J(k)$.

Let x be any follower at the end of some stage s. Let $i < 2$ such that $k^i_s(x)$ is defined. Then there are two possibilities:

(1) $M^i_s(x) = J\left(k^i_s(x)\right)$; we sometimes say that x *resides at its private box* at stage s; or

(2) $M^i_s(x) = B\left(k^i_s(x), \alpha\right)$ for some $\alpha \in {}^{\leqslant k+1}a(k)$; we let $\alpha^i_s(x)$ be this string α.

If $M^i_s(x) = J\left(k^i_s(x)\right)$ (x resides at its private box at stage s) then $i = \mathtt{top}_s(x) = 1$ and x did not move since it was appointed.

Let s be a *stage* ; let $i < 2$ and $k \geqslant c$. After it has been decided if any follower receives attention at *stage* s, and thus cancels weaker followers, the collection of followers which are alive at the end of the stage is determined. Based on this information, we define a string $\beta^i_s(k)$:

- Suppose that there is some follower y, alive at both the beginning and the end of stage s, which does not receive attention at stage s, and such that $\mathtt{top}_{s-1}(y) = i$, $k^i_{s-1}(y) = k$, and $M^i_{s-1}(y) \neq J(k)$. We then let y be the weakest such follower, and let $\beta^i_s(k) = \alpha^i_{s-1}(y)$.
- If there is no such follower y, then we let $\beta^i_s(k) = \langle\rangle$.

The main part of the proof will be the establishment of the following "bookkeeping" lemma:

Lemma 2.1. *For all s, $i < 2$ and $k \geqslant c$, $|\beta^i_s(k)| \leqslant k$. Furthermore, there is some $m < a(k)$ such that for all followers y which are alive at both the beginning and the end of stage s, if $k^i_{s-1}(y)$ is defined and equals k, and $M^i_{s-1}(y) \neq J(k)$, then $\alpha^i_{s-1}(x) \neq \beta^i_s(k)\hat{\ }m$. We let $m^i_s(k)$ be the least such m.*

Armed with Lemma 2.1, we can now give the full instructions for the construction.

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.

In the first case, we appoint a new, large follower x for P^e. We cancel all followers for requirements weaker that P^e. We set up x's parameters as follows:

- We define $k^0_s(x) = k^1_s(x) = e$.
- We let $\mathtt{top}_s(x) = 1$.
- We let $M^1_s(x) = J(e)$ (so x begins its life by residing in its private box).

- By Lemma 2.1, 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.

In the second case, 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 promote x; there are three cases.

1. 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.

2. If $\texttt{top}_{s-1}(x) = 1$ 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 $\texttt{top}_s(x) = 0$, 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 $t^0_s(x) = t^0_{s-1}(x)$.

3. If neither case (1) nor case (2) hold then we can have a "regular" promotion for x. Let $i = \texttt{top}_{s-1}(x)$.

- We let $\texttt{top}_s(x) = 1 - i$. 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 Lemma 2.1, 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 prove Lemma 2.1. The proof of this lemma 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 first establish some basic facts and notation. To begin, for $e \geqslant c$ and $s < \omega$, 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 $k^1_s(x)$ is undefined.

The following is immediate from the construction.

Lemma 2.2. *Let $x \in F_s$.*

(1) *For $i \in R_s(x)$, we have $k^i_s(x) \in [c, e]$.*
(2) $\texttt{top}_s(x) \in R_s(x)$.

(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 2.2, for any follower x, $k \geqslant c$ and $i < 2$, the value of $\mathtt{Par}^i_s(x)$ for stages s such that $x \in F_s$, $i \in R_s(x)$ and $k^i_s(x) = k$ does not depend on the choice of such a stage s. If there is any such stage, we let $\mathtt{Par}^i(k, x) = \big\langle k, M^i(k, x), t^i(k, x)\big\rangle$ be this value. 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 2.2. Let $i < 2$, and let $k \geqslant c$. Let $s < \omega$.

- 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 reside at 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 $s < \omega$ be a *stage*.

Lemma 2.3. *Let* $x \in KLG^i_s(k)$*. Let* $t = t^i_s(x) = t^i(k, x)$*. The stage* t *is 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 $y \in \overline{KG^i_t}(k)$, $\alpha^i(k,y) \neq \beta^i_t(k)\hat{}m^i_t(k)$.

Markers obey the priority ordering:

Lemma 2.4. *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 2.5. *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 2.4, 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 2.3 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 2.6. *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 2.3) $\gamma = \alpha^i(k,w)$ 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 2.6 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 2.7. *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 our main goal, Lemma 2.1, using this notation. The lemma says that $|\beta^i_s(k)| \leqslant k$ and that there is some $m < a(k)$ such that $\beta^i_s(k)\hat{\ }m \notin \overline{O^i_s}(k)$.

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 2.8. *Let s be a* stage. *Let $x \in F_{s-1}$, let $i = \mathtt{top}_{s-1}(x)$, and suppose that there is some $z \in M^i_{s-1}(x)$ such that $t^i_{s-1}(x) \notin T^i_s(z)$. 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 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 2.9. *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 2.6, $\alpha^i(k,y)$ 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 2.10. *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 2.8.

Now suppose that $x \in G^i_{s-1}(k)$. Let $z \in M^i(k,x)$. By Lemma 2.9, $\psi^i_{s-1}(z) = t^i(k,x)$. 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 2.8.

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 2.9, $\psi^i_{s-1}(z) = t^i(k,x)$. 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 2.11. *For all 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 2.7, 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 2.3). By Lemma 2.10, $t^i(k,x) \in T^i_s(z)$. By Lemma 2.4, 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 2.11 implies the first part of Lemma 2.1: for all s, $|\beta^i_s(k)| \leqslant k$. For if $\beta^i_s(k) \neq \langle\rangle$, then every nonempty initial segment of $\beta^i_s(k)$ equals $\alpha^i(k,x)$ for some $x \in K^i_s(k)$; this follows from the fact that $\beta^i_s(k) = \alpha^i\left(k, \max K^i_s(k)\right)$ and from Lemma 2.6.

Lemma 2.12. *For all 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 2.10, for all $x \in \overline{L^i_s(k)}$, $t^i(k,x) \in T^i_s(z)$; with Lemma 2.4 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, $\left|G^i_s(k)\right| < a(k)$.*

Proof. Let $x \in G^i_s(k)$. By Lemma 2.3, 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 Lemma 2.1.

Proof of Lemma 2.1. We have already seen that Lemma 2.11 implies the first part of the lemma: that $|\beta^i_s(k)| \leqslant k$. 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 $\beta^i_k(s) = \alpha^i(k,w)$ for $w = \max \overline{K^i_s}(k)$. But then $x \leqslant w$ and so (Lemma 2.7) 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 2.3).

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 2.11 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 enumerating 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. We claim that P^e does not appoint 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. At stage s_1, all followers weaker than the followers in H^e are cancelled, and so 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 $x \in H^e$, for 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 $x \in H^e$ is never realised. For if it is, then all followers in H^e are sometime realised, 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 $A^i \restriction v^i_s(x) = A^i_s \restriction v^i_s(x)$. 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 $i = \texttt{top}_{r-1}(x)$ 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 $i = \texttt{top}_r(x)$; if $i \neq \texttt{top}_s(x)$, then $i = \texttt{top}_r(x)$ where r is the next stage at which x receives attention. But then, the fact that x will not be permitted once i becomes $\texttt{top}(x)$, 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 $A^i \restriction v^i_s(x) = A^i_s \restriction v^i_s(x)$.*

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 is never realised, then Lemma 2.9 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 2.10, $v^i_s(x)\downarrow$. 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 $i = \texttt{top}_{r-1}(x)$, 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 $A^0 \upharpoonright v^0_s(x) = A^0_s \upharpoonright v^0_s(x)$.

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

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 2.11, $|K^0_\omega(c)| \leqslant c$. □

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 only 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 noncomputable, 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`. 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 and $\mathcal{N}^e = \bigcup_c \mathcal{N}^{e,c}$, $\mathcal{N} = \bigcup_e \mathcal{N}^e$, 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 assign 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, $\psi^\tau_r(z) = \psi^\tau_{s-1}(z)$ is an element of $T^\tau_s(z)$.

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 $\varphi^e_s(x) = 0$. The requirement P^e is *satisfied* at stage s if there is some $x \in E_s$ such that $\varphi^e_s(x) = 0$. 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 $k^\tau_s(x) \geqslant |\tau|$. This is the level at which x points.
- For all $\tau \in R_s(x)$, we define a box $M^\tau_s(x) \subset I^\tau(k^\tau_s(x))$.
- For all $\tau \in R_s(x)$, we define a value $t^\tau_s(x) < \omega$.

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 $z \in M^\tau_s(x)$ we have $t^\tau_s(x) \in T^\tau_s(z)$. In this case, we let

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

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

Let s be a stage at the beginning of which a follower x is alive, and suppose that s is a $\mathtt{top}_{s-1}(x)$-*stage*; let $\tau = \mathtt{top}_{s-1}(x)$. Let r be the previous *τ-stage* before s. We say that x is *permitted* at stage s if $v^i_r(x)\uparrow$ or if $W^e_s \restriction v^\tau_r(x) \neq W^e_r \restriction v^\tau_r(x)$, 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 $z \in M^\tau_s(x)$. 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 as it lies in its private box $\{z^\tau(\sigma(x))\}$, 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 is moved from private to 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. 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 $B^\tau(k,\alpha^\frown m)$ (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 resides in 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 $\alpha^\tau_s(x)$.

Suppose that s is a τ-*stage*. Once we know which followers are alive at the end of the stage, for $k \geqslant |\tau|$ we then define a string $\beta^\tau_s(k)$:

- If there is a follower y, alive at the beginning and the end 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 $\beta^\tau_s(k) = \langle\rangle$.

The main bookkeeping lemma is:

Lemma 3.1. *If $\tau \in \mathcal{N}$ and s is a τ-*stage*, then for all $k \geqslant |\tau|$, $|\beta^\tau_s(k)| \leqslant k$, and there is some $m < a(k)$ such that for all followers y which are alive ay both the beginning and the end of stage s, if $\tau \in R_s(y)$, $k^\tau_{s-1}(y) = k$, and $M^\tau_{s-1}(y) \not\subseteq J^\tau(k)$, then $\alpha^\tau_{s-1}(y) \neq \beta^\tau_s(k)^\frown m$. We let $m^\tau_s(k)$ be the least such m.*

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 $\tau^\frown\mathtt{fin}$ 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 $\tau^\frown\infty$ 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:

1. 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 $\sigma(x)$.

2. 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 $\mathtt{top}(x)$: we let $\mathtt{top}_s(x)$ be the longest $\rho \in R_s(x)$ for which $k^\rho_s(x)$ is maximal among the elements of $R_s(x)$.

3. 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 $\mathtt{top}(x)$ as we did in the second case: we let $\mathtt{top}_s(x)$ 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 $\mathtt{top}_s(x)$ 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 $\tau = \mathtt{top}_s(x)$, we let $M^\tau_s(x) = \{z^\tau(\sigma)\}$, the singleton subset of $J^\tau(|\sigma|)$ which is reserved for σ.
- For $\tau \in R_s(x)\backslash\{\mathtt{top}_s(x)\}$ we let $M^\tau_s(x) = B^\tau(|\sigma|, \beta^\tau_s(|\sigma|)\hat{\ }m^\tau_s(|\sigma|))$.

We then end the stage.

3.3. **Justification.** We need to prove Lemma 3.1 to show that the construction can be performed as prescribed. 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 $\sigma \in \mathcal{P}$, 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 $k^\tau_s(x) = k^\tau_r(x)$, then $M^\tau_s(x) = M^\tau_r(x)$ and $t^\tau_s(x) = t^\tau_r(x)$. Hence for $k = k^\tau_s(x)$ we let $M^\tau(k, x) = M^\tau_s(x)$ and $t^\tau(k, x) = t^\tau_s(x)$. If $M^\tau(k, x) \neq \{z^\tau(\sigma(x))\}$, then we let $\alpha^\tau(k, x) = \alpha^\tau_s(x)$.

Lemma 3.2. *Let $x \in F_s$, and let $k = k^{\mathtt{top}_s(x)}_s(x)$. Let $\tau \in R_s(x)$.*

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

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 second part of the lemma follows from the first. 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 $k =$

$k^\tau_{s-1}(x) = |\tau|$. 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 $k^\rho_{s-1}(x) = k-1$, which is smaller than $|\rho|$, which is impossible.

Now we prove the first part of the lemma, by induction on s.

If x is appointed at stage s, then for all $\tau \in R_s(x)$, $k^\tau_s(x) = k = |\sigma(x)|$, 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 $\overline{k} = k^{\overline{\tau}}_{s-1}(x)$. For all $\tau \in R_s(x)$ different from $\overline{\tau}$ we have $k^\tau_s(x) = k^\tau_{s-1}(x)$.

If $\overline{\tau} \notin R_s(x)$, then by the second part of the lemma, $\overline{\tau}$ is the longest node in $R_{s-1}(x)$. Hence for all $\tau \in R_s(x)$ we have $k^\tau_s(x) = \overline{k}$, 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 $k^{\overline{\tau}}_s(x) = \overline{k} - 1$. 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 $k^\tau_s(x) = \overline{k} - 1$; 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 $k^\tau_s(x) = \overline{k} - 1$, and if $\tau \subsetneq \overline{\tau}$ then $k^\tau_s(x) = \overline{k}$. We choose $\mathtt{top}_s(x)$ to be the immeidate predecessor of $\overline{\tau}$ in $R_s(x)$.

□

In particular, Lemma 3.2 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 $K^\tau_s(k)$ be the collection of followers $x \in F_s$ such that $\tau = \mathtt{top}_s(x)$, $k = k^\tau_s(x)$, and $M^\tau(k,x) \neq \{z^\tau(\sigma(x))\}$.
- We let $L^\tau_s(k)$ be the collection of followers $x \in F_s$ such that $\tau = \mathtt{top}_s(x)$, $k = k^\tau_s(x)$, and $M^\tau(k,x) = \{z^\tau(\sigma(x))\}$.
- We let $G^\tau_s(k)$ be the collection of followers $x \in F_s$ such that $\tau \in R_s(x)$, $k = k^\tau_s(x)$, but $\tau \neq \mathtt{top}_s(x)$.

Again we use the notation $KG^\tau_s(k)$ to denote the union $K^\tau_s(k) \cup G^\tau_s(k)$, and the notation $\overline{K^\tau_s}(k) = K^\tau_{s-1}(k) \cap K^\tau_s(k)$ etc. Again, $KLG^\tau_s(k)$ is the collection of followers $x \in F_s$ such that $\tau \in R_s(x)$ and $k = k^\tau_s(x)$, $KL^\tau_s(k)$ is the collection of followers $x \in KLG^\tau_s(k)$ such that $\tau = \mathtt{top}_s(x)$, and $KG^\tau_s(k)$ is the collection of followers $x \in KLG^\tau_s(k)$ such that $\alpha^\tau(k,x)$ is defined.

The following lemma translates the construction into this terminology:

Lemma 3.3. *Let $x \in KLG^\tau_s(k)$. 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 $L^\tau_t(k)$ if $\tau = \mathtt{top}_t(x)$, and into $G^\tau_t(k)$ if not.*
- *Otherwise, x is realised at stage t, and x is added to $G^\tau_t(k)$.*

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

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

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

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 2.4 to Lemma 2.11, including the first part of Lemma 3.1, that $|\beta^\tau_s(k)| \leqslant k$.

Lemma 3.4. *For all $\sigma \in \Theta^\tau(k)$, $|F^\sigma_s \cap L^\tau_s(k)| \leqslant k+1$.*

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 3.4 and the analogue of Lemma 2.11 tell us that $|KL^\tau_s(k)| \leqslant b(k)$.

Lemma 3.5. $|G^\tau_s(k)| < a(k)$.

Proof. Let $x \in G^\tau_s(k)$; let $\rho = \texttt{top}_s(x)$. By Lemma 3.2, $x \in KL^\rho_s(k)$ or $x \in KL^\rho_s(k+1)$. 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 $|KL^\rho_s(k)|$ and $|KL^\rho_s(k+1)|$ are bounded by $b(k)$ and $b(k+1)$ respectively, and so both bounded by $b(k+1)$. Hence $|G^\tau_s(k)| \leqslant 2^{k+2}b(k+1) = a(k) - 1$. □

The argument of the previous section now gives a proof of Lemma 3.1.

3.4. **Verification.**

Lemma 3.6. *Every follower receives attention only finitely many times.*

Proof. Identical to the proof of Lemma 2.15, using $k_s(x) = \sum_{\tau \in R_s(x)} k^\tau_s(x)$. □

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 3.7. *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 $KLG^\tau_s = \bigcup_{k \geqslant |\tau|} KLG^\tau_s(k)$. If $x \in KLG^\tau_s \backslash KLG^\tau_{s-1}$ then some $\sigma \supseteq \tau\hat{\ }\infty$ is accessible at stage s. This means that $s < s_0$. That is, no new followers are added to KLG^τ_s after stage s_0. By Lemma 3.6, each follower in $KLG^\tau_{s_0}$ receives attention finitely many times. We reach a contradiction. □

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

Lemma 3.8. *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 3.9. *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 3.7, it suffices to show that there are infinitely many τ-*stages*. This follows from the fact that T^τ traces ψ^τ. □

Lemma 3.10. *E is not computable.*

Proof. We need to show that every requirement P^e is met. By Lemma 3.8, 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 3.9, 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 3.9, $\tau\hat{\ }\infty$ lies on the true path; there are infinitely many τ-*stages*.

Lemma 3.11. *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 3.12. *Let s be a τ-stage. Let $x \in F_s$ such that $\tau \in R_s(x)$ and suppose that $v^\tau_s(x)\downarrow$. Suppose that $W^e \restriction v^\tau_s(x) = W^e_s \restriction v^\tau_s(x)$. Then $x \notin E$.*

Proof. Identical to the proof of Lemma 2.17, using Lemma 3.11. □

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 3.13. *Let $x \in H$ and let $\tau \in R_\omega(x)$. There is some τ-*stage *s such that $x \in F_s$, $v^\tau_s(x)\downarrow$ and $W^e \restriction v^\tau_s(x) = W^e_s \restriction v^\tau_s(x)$.*

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

Lemma 3.14. *$E \leqslant_{\mathrm{T}} W^e$.*

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
- $\tau \in R_s(x)$, $v^\tau_s(x)\downarrow$ and $W^e \restriction v^\tau_s(x) = W^e_s \restriction v^\tau_s(x)$.

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 3.8.

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 3.13 ensures that a stage s as above is found. Otherwise, let $\rho = \texttt{top}_\omega(x)$. Let $r+1$ be the stage at which τ is removed from $R(x)$; so $k_r^\tau(x) = |\tau|$. Lemma 3.2 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^0_1 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 mathematics 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/