

LOW₂ COMPUTABLY ENUMERABLE SETS HAVE HYPERHYPERSIMPLE SUPERSETS

PETER CHOLAK, RODNEY DOWNEY AND NOAM GREENBERG

ABSTRACT. A longstanding question is to characterize the lattice of supersets (modulo finite sets), $\mathcal{L}^*(A)$, of a low₂ computably enumerable (c.e.) set. The conjecture is that $\mathcal{L}^*(A) \cong \mathcal{E}^*$. In spite of claims in the literature, this longstanding question/conjecture remains open. We contribute to this problem by solving one of the main test cases. We show that if c.e. A is low₂ then A has an atomless hyperhypersimple superset. In fact, if A is c.e. and low₂, then for any Σ_3 -Boolean algebra B there is some c.e. $H \supseteq A$ such that $\mathcal{L}^*(H) \cong B$.

1. INTRODUCTION

The concern of this paper is the interplay of two fundamental structures arising in classical computability, the lattice of computably enumerable sets and Turing reducibility \leq_T . Turing [Tur39] defined $A \leq_T B$ as the most general notion of reducibility between problems coded as sets of non-negative integers. Reducibilities give rise to partial orderings which measure relative computational complexity. In the classic paper [Pos44], Post suggested that the study of the lattice of computably (recursively) enumerable (c.e.) sets was fundamental in computability theory. In the words of Soare [Soa87, p. viii]

“Post [Pos44] stripped away the formalism associated with the development of recursive function theory in the 1930’s and revealed in a clear informal way the essential properties of r.e. sets and their role in Gödel’s incompleteness theorem”.

Computably enumerable sets are the halting sets of Turing machines, i.e., the domains of partial computable functions. They thus represent natural “semi-decidable” problems such as instances of Hilbert’s 10th problem, the Entscheidungsproblem, the Post Correspondence Problem, sets of consequences of formal systems, and many others.

The interplay of these two basic objects, Turing reducibility and c.e. sets, has a long and rich history. The computably enumerable sets under union and intersection form a lattice, denoted by \mathcal{E} , and a common object of study is \mathcal{E}^* , which is \mathcal{E} modulo the congruence $=^*$, where $A =^* B$ means that the symmetric difference of A and B is finite. The Turing degrees of c.e. sets form an upper semilattice, denoted by \mathcal{R} . Ever since the groundbreaking paper of Post, there has been a persistent intuition that structural properties of computably enumerable sets have reflections in their degrees, and vice versa. In particular, *definability* in \mathcal{E} should be linked with *information content* as measured by Turing reducibility.

The simplest possible illustration of this is the fact that the complemented members of \mathcal{E} are exactly the members of $\mathbf{0}$, the degree of the computable sets.

The authors thank the Marsden Fund of New Zealand.

One of the main avenues of attack has been to link properties of c.e. sets with the jump operator, $A' = \{e \mid \Phi_e^A(e) \downarrow\}$, the halting problem *relative to* A . The jump operator gives one way of understanding information content of members of \mathcal{R} . A deep example is Martin's result [Mar66] that the Turing degrees of maximal sets are exactly the high c.e. Turing degrees. Here a c.e. set A is *high* if $\emptyset'' \equiv_T A'$, i.e., if the jump operator does not distinguish between A and the halting problem \emptyset' . A coinfinite c.e. set A is *maximal* if it represents a co-atom in \mathcal{E}^* , that is, if for every c.e. set $W \supseteq A$, either $A =^* W$ or $A =^* \mathbb{N}$.

On the other hand, we call A *low* if $A' \equiv_T \emptyset'$, that is, if the jump operator does not distinguish between A and the computable sets. Such sets have relatively little information content, so we would guess that low sets should perhaps share some properties with the computable sets.

This intuition has seen many realizations in the study of the Turing degrees of low c.e. sets. One such example is Robinson's [Rob71] result that if we have computably enumerable sets $B \leq_T A$ with B low, then we can split $A = A_1 \sqcup A_2$ as two c.e. sets with $A_1 \oplus B$ Turing incomparable with $A_2 \oplus B$. This result shows that splitting and density can be combined above low c.e. degrees, something that, famously, Lachlan [Lac76] showed fails in general.

How is this intuition reflected in the properties of low c.e. sets in \mathcal{E}^* ? Soare [Soa82] showed that if A is low then $\mathcal{L}^*(A) \cong \mathcal{E}^*$. Here $\mathcal{L}^*(A)$ denotes the lattice of c.e. supersets of A modulo $=^*$. Thus it is impossible to distinguish A from \emptyset using properties of its lattice of supersets. We remark that Soare proved that the isomorphism is *effective* in that there are computable function f and g with $W_e \mapsto W_{f(e)}$ (from $\mathcal{L}^*(A)$ to \mathcal{E}^*) and $W_e \mapsto W_{g(e)}$ for the return map, inducing the isomorphism.

Note that Martin's Theorem says that if a c.e. degree is high then then it contains a maximal set A . If A is maximal then $\mathcal{L}^*(A)$ is the two-element lattice — as far as possible from \mathcal{E}^* . It is natural to wonder what level of computational power for degrees stop c.e. sets with those degrees from resembling computable sets. A natural demarcation seems to be at low_2 . Recall that a set A is low_2 if the *double* jump does not distinguish between A and the computable sets: $A'' \equiv_T \emptyset''$.¹ Shoenfield [Sho76] proved that if a c.e. Turing degree is not low_2 , then it contains a c.e. set B that has no maximal superset. Hence, in particular, $\mathcal{L}^*(B) \not\cong \mathcal{E}^*$. It follows that if there is a collection \mathcal{J} of degrees characterized by their jumps, such that $\text{deg}(A) \in \mathcal{J}$ implies $\mathcal{L}^*(A) \cong \mathcal{E}^*$, then \mathcal{J} must be a subclass of the low_2 degrees.

Maass [Maa83] extended Soare's result to prove that for a c.e. set A , $\mathcal{L}^*(A)$ is effectively isomorphic to \mathcal{E}^* if and only if A is $\text{semilow}_{1.5}$. Here a c.e. set A is called $\text{semilow}_{1.5}$ if $\{e : W_e \subseteq^* A\}$ is Σ_2 . Being $\text{semilow}_{1.5}$ is a "pointwise" variation of lowness. It is known that a c.e. degree that is not low contains a c.e. set which is not $\text{semilow}_{1.5}$. In particular, there are low_2 sets which are not $\text{semilow}_{1.5}$ (Downey, Jockusch and Schupp [DJS13, Theorem 1.5]).

The following conjecture has been around for some time.

Conjecture 1.1 (Soare and others). *If A is a coinfinite low_2 c.e. set then $\mathcal{L}^*(A) \cong \mathcal{E}^*$.*

¹Indeed the reader should recall that, more generally, a set A is low_n if $A^{(n)}$, the n^{th} jump of A , is as low as possible: $A^{(n)} = \emptyset^{(n)}$, equivalently, $\Delta_{n+1} = \Delta_{n+1}^A$; and a Δ_2 set A is high_n if its n^{th} jump is as high as possible: $A^{(n)} = \emptyset^{(n+1)}$, equivalently, $\Delta_{n+2} = \Delta_{n+1}^A$.

This conjecture is stated as an open question in Soare [Soa87]. We remark that Conjecture 1.1 has been claimed as a theorem in several places, notably as being due to Harrington, Lachlan, Maass and Soare as stated in Harrington-Soare [HS96, Theorem 1.8], as well as in lectures by Soare in the late 1990's. But despite the best efforts of computability-theorists in the last 30 or so years, no proof has appeared, so it is surely an open question by this stage.

Notice that for this conjecture to hold, we must need some other method of constructing the isomorphism than an effective isomorphism. Likely if Conjecture 1.1 is true, then methods along the lines of the powerful Δ_3 methods introduced by Soare and Harrington [HS96] or Cholak [Cho95] will likely be needed. Cholak [Cho95] gave some partial evidence for the validity of the conjecture by showing that every semilow₂ c.e. set with the *outer splitting property* has $\mathcal{L}^*(A) \cong \mathcal{E}^*$. It is not important for our story what these properties are, save to say that they give weaker guessing methods than semilow_{1.5}-ness, but which are sufficient given the Δ_3 isomorphism method. In this paper we make a modest contribution supporting Conjecture 1.1. We will soon explain what we mean by guessing, and how our work fits into the programme of establishing the conjecture.

Lachlan [Lac68a] proved that if A is low₂ then A has a maximal superset (i.e., $\mathcal{L}^*(A)$ contains a co-atom). This is currently the strongest general result about all low₂ c.e. sets. It is likely not too difficult to modify Lachlan's technique to extend this result to k -quasimaximal sets, c.e. sets which are the intersections of exactly k maximal sets, by simultaneously constructing k maximal supersets M_1, \dots, M_k of A such that $M_i \neq^* M_j$ for $i \neq j$. If Q is k -quasimaximal then $\mathcal{L}^*(Q)$ is a k -atom boolean algebra.

Lachlan [Lac68b] proved that $\mathcal{L}^*(A)$ is a Boolean algebra if and only if it is hyperhypersimple. Hyperhypersimple sets were defined, if not constructed, by Post [Pos44]. Recall that a coinfinite c.e. set A is called *simple* if its complement $A^c = \mathbb{N} \setminus A$ contains no infinite c.e. subsets. A much stronger property is that A is *hyperhypersimple*, which is defined as there being no collection of infinitely many finite, pairwise disjoint, uniformly c.e. sets, such that each element of the collection intersects the complement of A . However, Lachlan's characterisation in terms of Boolean algebras is used more often. A hyperhypersimple set A is *atomless* if $\mathcal{L}^*(A)$ is the atomless Boolean algebra. In a talk in 2006, Cholak pointed out the next test case for the low₂ conjecture: atomless hyperhypersimple sets. In this paper we affirm Cholak's conjecture.

Theorem 1.2. *Every coinfinite low₂ c.e. set has an atomless hyperhypersimple superset.*

We now try to explain why this is an important test case for the full low₂ conjecture. By necessity, this explanation is somewhat technical as it revolves around issues of Δ_3 guessing and arguments akin to those of the deep paper [HS96] by Harrington and Soare.

The reader might recall Friedberg's [Fri58] construction of a maximal set M . For each e , we need to ensure that if $W_e \supseteq M$, then either $W_e =^* M$ or $W_e =^* \mathbb{N}$, whilst still keeping M coinfinite. At each stage s we will have listed in order the elements of the complement $M_s^c = \{b_{i,s} \mid i \in \omega\}$. To make M coinfinite, we make sure that for each e , $\lim_s b_{e,s} = b_e$ exists. For the maximality requirement, the idea is to make, for all $n \geq e$, the n^{th} member of the complement of M a member of W_e , as follows. Assuming that $b_{e,s}, \dots, b_{n-1,s} \in W_{e,s}$ but $b_{n,s} \notin W_{e,s-1}$, if we see

some $x = b_{m,s} \in W_{e,s}$ for $m \geq n$, we make $x = b_{n,s+1}$ by enumerating the elements $b_{n,s}, \dots, b_{m-1,s}$ into M_{s+1} , causing $b_{n,s+1} \in W_e$. If this happens for each $n \geq e$ then $W_e \cup M =^* \mathbb{N}$. On the other hand, if we fail to keep finding such $b_{m,s}$, from some point n onwards, then $W_e \cup M =^* M$.

Of course, we have to worry about differing requirements putting things into M since W_e might want to make $b_{n,s} \in W_e$, and some other $W_{e'}$ might want to put this into M to cause $b_{n,t} \in W_{e'}$, so we need some way to reconcile such conflicting requests. This is done using Friedberg’s priority method. In modern terminology, if $e' > e$, then the requirement dealing with $W_{e'}$ will guess the eventual behaviour of W_e , and therefore be able to align with it: either wait for elements of M^c to enter W_e , before attempting to find elements of the complement in $W_{e'}$; or only start acting at a stage after which no more elements of M^c enter W_e . To implement this, Friedberg’s brilliant idea was using *e-states*.² The *e-state* of b at stage s is the binary string that records the indices $d \leq e$ such that $b \in W_{d,s}$. Friedberg’s idea is to put, for each $n \geq e$, $b_{n,s}$ into the lexicographically *maximized* *e-state*. This means that almost all of the complement of M will be in the same *e-state*, and hence be in W_e or out of W_e . In our constructions below, we use the modern tree-of-strategies terminology, rather than explicitly using *e-states*. However, the notion of *e-states* appears necessary when constructing isomorphisms of lattices of c.e. sets, and so will be useful for the current discussion.

Suppose that we try to emulate Friedberg’s method to construct a maximal $M \supset A$, where the opponent is controlling the c.e. set A . It is within the opponent’s power to take elements from the complement M_s^c such as $b_{i,s}$, and declare that they are in M , since they are in A_{s+1} , and $M \supseteq A$. The danger in implementing the strategy above for dealing with W_e is that we could believe that we see infinitely many elements of M^c in W_e , and keep dumping elements not yet seen in W_e into M , but then the elements of W_e go into A . This would result in M being cofinite. We should think of the elements of A as “phantom elements”, ones that shouldn’t really be considered, except that we cannot know this in advance. The way around this is for the requirement to *guess* whether there are infinitely many numbers that are in W_e and not in A , i.e., if $W_e \setminus A$ is infinite. This statement is $\Pi_2(A)$; since A is low₂, this means that this statement is Δ_3 .

Lachlan’s proof in [Lac68a] (see also [Soa87, Theorem XI.5.1]³) is a reasonably delicate construction that uses a version of Δ_3 guessing. Lachlan’s original proof seemed ad hoc and combinatorial. As part of our paper we give a new thematic proof of Lachlan’s Theorem based around Δ_3 guessing on an ω -branching priority tree. As both the “yes” and “no” answer to the question whether $W_e \setminus A$ is infinite are Σ_3 (as $\Delta_3 = \Sigma_3 \cap \Pi_3$), the outcomes are pairs (“yes”, n) and (“no”, n) where n is the witness for the Σ_3 predicate holding. When considering more than one requirement, we are guessing the answers for a Boolean combination of $\Pi_2(A)$ questions; but these too will be Δ_3 , so will be subject to the same guessing procedure. Δ_3 priority arguments are much less common in computability theory than the usual infinite injury arguments (such as the Thickness Lemma, or the Minimal Pair argument) which can be performed on finitely branching trees. Harrington and Soare [HS96], Downey and Greenberg [DG20], Shore and Slaman [SS90, SS93] are some examples of Δ_3 arguments.

²Thus, in effect, the *e-state* method is a forerunner of infinite injury arguments.

³This is the notorious “blondes and brunettes” construction in Soare.

The method of Δ_3 guessing does not appear to be sufficient for results stronger than the construction of a maximal superset. Suppose that we are trying to show that $\mathcal{L}^*(A) \cong \mathcal{E}^*$. For each e and each e -state σ , we need to ensure that infinitely many elements of \mathbb{N} have e -state σ (with respect to the list of c.e. subsets of \mathbb{N}) if and only if infinitely many elements of A^c have e -state σ , with respect to a listing of the c.e. supersets of A . The e -states correspond to measures of intersections and non-intersections of c.e. sets.

The method of Δ_3 guessing will allow us to correctly guess which e -states we should try to fill. But now we have the more complicated task of actually finding potential elements and enumerating them into the correct sets, so that we can match e -states as required. The difference between the maximal set construction and the more general construction is that in the former, for each e , there will be exactly one e -state that we need to populate. So there, the only difficulty is in guessing which e -state will be populated; once this is decided, all elements of the complement of M will be enumerated into the same sets (as usual, except for finitely many). If there are more than one e -state to populate, say σ and τ , then given an element b that at a stage s seems to be in the complement of the set we are building, we need to decide whether to enumerate b into sets so as to make the e -state of b either σ or τ . We need to ensure that both σ and τ will be populated by infinitely many “true” elements, elements that are not in A . The question is whether given such b at stage s , should we believe that this is the true situation, and, moreover, what do we do if we act on false beliefs?

The key to all proofs we mentioned (Soare, Maass, Cholak, etc) where $\mathcal{L}^*(A)$ is shown to be isomorphic to \mathcal{E}^* is some kind of guessing procedure to understand when elements seem to be truly outside A . If A is low, semilow, or even semilow_{1.5} we have a “pointwise” testing process where we can guess whether individual elements are in A^c , and get the answer *more-or-less immediately*. For example, suppose that A is low. Then using, for example, the Robinson trick, we can ask A' , and hence \emptyset' , whether some $z \in A_s^c$ (in some desirable e -state) is actually outside A , and must eventually get a true “yes” if there are infinitely many such potential z . In his proof that semilow_{1.5} c.e. sets have lattices of supersets effectively automorphic with \mathcal{E}^* ([Maa83]), Maass points out that indeed, it is not enough to know that there are infinitely many elements in some state, but we also need to capture them. Maass’s outer splitting property, mentioned above, is an elaboration on the low guessing technique, that works if A is semilow_{1.5}.

The key problem appears to be that of *splitting* e -states. For some e -state σ , we have somehow guaranteed that infinitely many true elements (elements outside A) have e -state σ . Suppose that Δ_3 guessing tells us that both σ^0 and σ^1 are $(e+1)$ -states that need “filling”. In the construction, we are given an infinite set E of elements in state σ , and we know that infinitely many of these are outside A . The problem is to split E into two sets E_0 and E_1 , while ensuring that both $E_0 \setminus A$ and $E_1 \setminus A$ are infinite. We would then enumerate the elements of E_1 into the $(e+1)^{\text{th}}$ c.e. superset of A , making them have $(e+1)$ -state σ^1 , while keeping the elements of E_0 outside that set, making them have $(e+1)$ -state σ^0 . This is precisely what is required in the construction of an atomless hyperhypersimple superset. As Cholak pointed out,⁴ the dynamics of Lachlan’s construction do not seem to be modifiable to obtain this.

⁴As did Maass in unpublished handwritten notes from a seminar.

In our main construction in Section 3, we introduce a novel method for performing this splitting. This method does not rely solely on Δ_3 guessing; we also use domination properties of low_2 sets, namely, the fact that if A is low_2 , then there is a \emptyset' -computable function that dominates all functions computable from A .

To end this discussion, we should mention why our new technique does not seem to immediately solve the original problem of showing that $\mathcal{L}^*(A) \cong \mathcal{E}^*$. The issue is very delicate, and as is often the case, relies on discrepancies in timing. When constructing an atomless hyperhypersimple superset H , we are, in some sense, controlling the supersets of H , which means that all e -states have equal status (this is by necessity an imprecise simplification). When constructing an isomorphism, the opponent has within their power to shift elements from one e state to another. There is a difference between what are colloquially known as “low” and “high” e -states. The opponent can move elements from low to high e -states. Unlike the low guessing construction, or the construction relying on the outer splitting property, our splitting method is not immediate: we have to wait an unknown amount of time until we get a certification that certain elements are outside A , and for finitely many elements, such certification may never happen. While we are waiting for certification, the opponent will shift elements outside a desirable low state. With our new splitting method, we are thus able to split high states, but not low states. In the hyperhypersimple construction, all states are high.

In addition to Lachlan’s characterisation of hyperhypersimple sets, he also classified the isomorphism types of the resulting Boolean algebras: he showed that the Boolean algebras $\mathcal{L}^*(A)$ for hyperhypersimple sets A are precisely the Σ_3 -Boolean algebras — quotients of the computable, atomless Boolean algebra by Σ_3 ideals. We show in this paper that Theorem 1.2 can be extended to obtain all such algebras as the lattice of supersets of a superset of any given low_2 c.e. set.

Theorem 1.3. *Suppose that A is a coinfinite, low_2 c.e. set. Let B be a Σ_3 boolean algebra. Then there is a c.e. superset $H \supseteq A$ with $\mathcal{L}^*(H) \cong B$.*

2. MAXIMAL SUPERSETS

As mentioned above, in order to present our proof of Theorem 1.2, it would be useful to first give a “modern” or at least “thematic” proof of Lachlan’s theorem, using Δ_3 guessing on a priority tree:

Theorem 2.1 (Lachlan [Lac68a]). *Every low_2 coinfinite c.e. set has a maximal superset.*

Discussion. We are given a coinfinite, low_2 c.e. set A ; we enumerate a maximal c.e. set $M \supseteq A$.

This proof follows, to a certain extent, the Δ_3 automorphism machinery of Harrington and Soare [HS96]. The construction is performed on a tree of strategies, similarly to many infinite injury priority arguments. Nodes on the tree represent guesses about the eventual behaviour of some aspects of the construction, and use their guesses to meet a requirement that they are assigned to. In the current construction, nodes of length $e + 1$ make guesses about how the e^{th} c.e. set W_e interacts with the construction, and attempt to meet the e^{th} maximality requirement: either $W_e \cup M =^* \mathbb{N}$, or $W_e \cup M =^* M$. We will identify a *true path*, the path of nodes whose guesses are correct; nodes on the true path will be successful in meeting their requirements.

In some ways, though, the usage of the tree of strategies is quite different to most constructions. For one, at a stage s of the construction, we do not define a path of “accessible” nodes, those whose guesses appear to be correct at that stage. Further, there will not be much explicit interaction between strategies: they will not impose restraint, or cause initialisation of other strategies. We will not use the terminology of relative priority.

More importantly, we use the tree as the hardware of a *pinball machine*. We view the priority tree as growing downwards; so the root of the tree is the “top” node. During the construction, we place some *balls* at the root of the tree. These balls represent numbers that at a stage s of the construction are not in A_s (the stage s approximation to the given c.e. set A). At any given stage, there will be only finitely many balls on the machine, but we will ensure that every element of the complement A^c of A is placed on the machine at some stage. Once placed at the root, we allow balls to move between nodes on the tree. Some balls may be enumerated into M , at which time they are removed from the machine. This includes all the elements of A , since $A \subseteq M$. We will ensure that balls that are never enumerated into M only move during finitely many stages of the construction, so they eventually arrive at a permanent “resting place”. The main mechanics of the construction are the determination of which balls move, where they move to, and which balls are removed from the machine.

We remark that the term “pinball machine” was earlier used by Lerman, to organise priority arguments (often used for embedding lattices into the c.e. degrees). The Harrington-Soare pinball machine, that we use here, is different in that the balls move from the root to other nodes, which is an opposite direction to Lerman’s machines.

As mentioned in the introduction, another aspect of the Harrington-Soare machinery is the employment of Δ_3 -guessing. This is the main way that we use, in this construction, the assumption that A is low₂. In a typical Π_2 argument (such as the minimal pair construction), during the construction we guess the outcome of Π_2 questions based on what we see at each stage. In the current construction, we ask more complicated questions, namely, $\Pi_2(A)$ questions, and we use the fact that $\Pi_2(A) \subseteq \Delta_3$. Rather than observing the construction and making guesses accordingly, we are given an approximation to the answer to a Δ_3 statement, that may or may not be aligned with what is measured at a given stage; we just know that in the limit, the approximation will give us the correct answer. As we shall detail below, we will rely on the recursion theorem to ensure that we indeed approximate the correct answers. The guessing process for Δ_3 facts is more complicated than that of Π_2 facts. Namely, to guess membership in a Δ_3 set S , we need to guess an answer ($n \in S$ or $n \notin S$), and to guess an existential witness for the Σ_3 predicate being guessed. This implies that the tree of strategies needs to be infinite-branching.

The requirements. Let us consider now the goals of the construction. As discussed, the balls that are never removed from the machine will be precisely the elements of M^c . Each such ball will eventually settle as a “resident” of some node β on the tree. For a node α on the tree, we will let $Y(\alpha)$ denote the collection of balls which are permanent residents of nodes $\beta \succcurlyeq \alpha$. The rules about ball movement will ensure that $Y(\alpha)$ is a d.c.e. set. To enter $Y(\alpha)$, a ball x will need to first pass through α , i.e., be a resident of α at some stage. It may later move to extensions of α . It is possible that it will be removed from $Y(\alpha)$ by being *pulled* by some node, that

lies to the left of α on the tree. In any case, balls can be removed from $Y(\alpha)$ by enumerating them into M . We will ensure, however, that $Y(\alpha) \cup M$ is in fact a c.e. set, though not uniformly in α . This is because either only finitely many balls outside M will ever enter $Y(\alpha)$, or finitely many balls outside M ever leave $Y(\alpha)$, or $Y(\alpha) = \emptyset$.

We will ensure the following:

- (a) For each node β on the true path, $Y(\beta) =^* M^c$. That is, all but finitely many elements of M^c will have passed through β at some stage and settled at β or some extension of β , i.e., below β (recall that our trees grow downwards).
- (b) For each e , if β is the node on the true path of length $e + 1$, then $Y(\beta)$ is either almost contained in, or almost disjoint from, W_e .
- (c) We also need to ensure that M is coinfinite.

For (a), we will ensure that only finitely many balls have permanent residence to the left of β , to the right of β , or above β .

For (b), we will ensure that either all balls that ever pass through β are already seen to be in W_e when they do, or that we know that only finitely many elements of W_e will ever be available to pass through β . In the first case, the node β will ensure that all but finitely many balls outside W_e will be enumerated into M (we will say that they are *eliminated* from the machine).

For (c), we will ensure that infinitely many nodes β on the true path hold on to balls and ensure that they are not eliminated, i.e., keep them out of M . Similar action will ensure that balls indeed only move finitely much, i.e., they do eventually settle at some node.

Setup. We now go into the details.

The tree. We let the tree of strategies be the collection of all sequences of the symbols \mathbf{fin}_n and ∞_n . These symbols are called *outcomes*. We write $\alpha \preceq \beta$ to indicate that α is a prefix of β .

In addition to extension, we will use the *Kleene-Brouwer* ordering on the tree. To define this we fix an ordering of the outcomes, say

$$\infty_0 < \mathbf{fin}_0 < \infty_1 < \mathbf{fin}_1 < \dots$$

We say that α lies to the left of β , and write $\alpha <_L \beta$, if α and β are incomparable (neither is a prefix of the other), and $\alpha(k) < \beta(k)$ for the least k such that $\alpha(k) \neq \beta(k)$. We write $\alpha \preceq \beta$ if either $\alpha <_L \beta$ or $\beta \preceq \alpha$. As this is the only linear ordering of strategies that we will use, we use the notation $\alpha \preceq \beta$ rather than $\alpha \preceq_{KB} \beta$.

We let λ denote the root of the tree (the empty sequence).

Notation for the pinball machine. As discussed, at each stage, finitely many balls x will reside at some nodes of the machine. We let

$$S(\alpha)_s$$

denote the collection of x that reside at the node α at the *beginning* of stage s . We let $Y(\alpha)_s$ be the collection of all balls that at the beginning of stage s reside at α or below α , i.e., at some extension of α :

$$Y(\alpha)_s = \bigcup_{\beta \succeq \alpha} S(\beta)_s.$$

So $Y(\lambda)_s$ is the collection of balls that are on the machine at the beginning of stage s .

True stage, and delayed, enumerations. Since the given set A is low₂, it is not high. C.e. sets that are not high have “true stage” enumerations with respect to any given computable growth rate: for any computable function f , there is a computable enumeration (A_r) ⁵ of A such that for infinitely many stages r , the first $f(r)$ many elements of the stage r complement A_r^c are “correct”: they are elements of the complement A^c . See [Soa87, Lem. XI.1.6].⁶

We fix an enumeration (\tilde{A}_r) of A , that is true with respect to the function $f(r) = r^2$. During the construction, though, it will be useful to *delay* this enumeration. We will define, during the construction, an enumeration (A_s) of A , such that for all s there is some r such that $A_s = \tilde{A}_r$. We will write $n(s) = r$.

Let s be a stage, and suppose that $n(s) = r$. We will let the construction run: balls will move around, or be enumerated into M , and it will be useful to keep track of these actions as happening during several stages: $s, s + 1, s + 2$, and so on. While we are doing this, we keep our enumeration of A fixed: that is, $\tilde{A}_r = A_s = A_{s+1} = A_{s+2} = \dots$. After finitely many stages, we will reach a stage $t \geq s$ at which we decide that there is nothing we want to do. We then declare t to be a “new balls” stage, and set $A_{t+1} = \tilde{A}_{r+1}$,⁷ i.e., set $n(t + 1) = r + 1$.

Thus, the function $s \mapsto n(s)$ will be weakly increasing and onto; we will have $n(s + 1) \neq n(s)$ exactly when s is a new balls stage, in which case $n(s + 1) = n(s) + 1$. What is important is that the enumeration (\tilde{A}_r) is given to us, whereas the enumeration (A_s) is defined by us during the construction: based on how the construction develops, we decide whether to declare a “new balls” stage, i.e., when to increase n by 1 and thus enumerate more numbers into A . We start with $n(0) = 0$. Observe that for any stage s , $n(s)$ is the number of new balls stages $t < s$. We will ensure that there are infinitely many new balls stages, so that indeed $\bigcup_s A_s = A$.

Definition 2.2. We let

$$Q_s = \{\text{the smallest } n(s)^2 \text{ many elements of } A_s^c\}.$$

A stage s is *A-true* if $Q_s \subseteq A^c$.

Thus, if $s < t$ are successive new balls stages, then $Q_{s+1} = Q_{s+2} = \dots = Q_t$, and $s + 1$ is *A-true* if and only if all the stages $s + 1, s + 2, \dots, t$ are *A-true*. A stage s is *A-true* if and only if the stage $r = n(s)$ is true for the enumeration (\tilde{A}_r) with respect to $r \mapsto r^2$. By assumption, there are infinitely many such r . As we will ensure that $s \mapsto n(s)$ is onto, there will be infinitely many *A-true* stages s .

We will ensure that at every stage s , $Y(\lambda)_s \setminus M_s = Q_s$. So at a new balls stage s , we will place all the elements of Q_{s+1} that have not yet been put on the machine, at the root. As stages go by, some of these elements may be enumerated into M .

⁵Here the notation (A_r) refers to the infinite sequence of finite sets A_0, A_1, \dots that form the enumeration of A .

⁶Note that the term “true stages” is sometimes used to refer to the Dekker “nondeficiency” stages; we will not use these.

⁷Recall that the index s denotes the objects as they are at the *beginning* of stage s . During stage s we perform some actions, resulting in the objects indexed by $s + 1$. Thus, if during stage t we place new balls on the machine, these balls will be elements of $Y(\lambda)_{t+1}$, not of $Y(\lambda)_t$.

The reason for using true stages for the function $f(r) = r^2$ is the following. If s is a new balls stage then

$$|Q_{s+1}| = |Q_s| + 2n(s) + 1 > |Q_s| + n(s).$$

Thus, at a new balls stage s we will be adding more than $n(s)$ -many new balls to the root of the machine; and we will use the fact that $n(s) \rightarrow \infty$. As time goes by, we will be getting balls at the root in increasing number, and infinitely often, all of these balls will be A -correct.

Δ_3 *guessing*. By assumption, A is low_2 , i.e., $\Pi_2(A) \subseteq \Delta_3$. Thus, any Boolean combination of $\Pi_2(A)$ statements is equivalent to a Σ_3 statement. Since there is a universal $\Pi_2(A)$ set, this is effective: given ψ , a Boolean combination of $\Pi_2(A)$ statements, we can effectively find a Σ_3 statement χ such that ψ holds if and only if χ holds.

In turn, from χ we can effectively produce a uniformly computable collection of nondecreasing sequences $\bar{\ell}(\psi, n) = (\ell(\psi, n)_s)$ for $n \in \mathbb{N}$ such that:

- ψ holds if and only if for some n , $\bar{\ell}(\psi, n)$ is unbounded.

To see this, write χ as $\exists n \forall x \exists y \theta(n, x, y)$. Then we let $\ell(\psi, n)_s$ be the greatest $x < s$ such that for all $x' \leq x$, there is some $y < s$ such that $\theta(n, x', y)$ holds.

Shortly, for each node α on the tree of strategies, we will formulate an “ α question” $\psi(\alpha)$, which will be a $\Pi_2(A)$ statement. In other words, the collection of nodes α such that $\psi(\alpha)$ holds is $\Pi_2(A)$. Using the transformation above for the statements $\psi(\alpha)$ and $\neg\psi(\alpha)$ (both are Boolean combinations of $\Pi_2(A)$ statements), we obtain families of sequences $\bar{\ell}(\psi(\alpha), n)$ and $\bar{\ell}(\neg\psi(\alpha), n)$. By the recursion theorem, we know a computable index for the construction, and so can have access to these families of sequences during the construction.⁸

The α question.

Definition 2.3. Let α be a node. The statement $\psi(\alpha)$ is:

For every $k \in \mathbb{N}$ there is an A -true stage s for which

$$|Y(\alpha)_s \cap W_{|\alpha|,s}| \geq k.$$

As just discussed, during the construction we have access to the sequences $\bar{\ell}(\psi(\alpha), n)$ and $\bar{\ell}(\neg\psi(\alpha), n)$. Now for each node β we define a (single) sequence $\bar{\ell}(\beta)$ by induction on the length $|\beta|$:

- $\ell(\lambda)_s = s$;
- $\ell(\beta \hat{\circ} \infty)_s = \min\{\ell(\beta)_s, \ell(\psi(\beta), n)_s\}$;
- $\ell(\beta \hat{\text{fin}})_s = \min\{\ell(\beta)_s, \ell(\neg\psi(\beta), n)_s\}$.

Thus, via the sequences $\bar{\ell}(\gamma)$, the children γ of β together try to answer the β question.

⁸For the reader who may less comfortable with such a use of the recursion theorem, we provide a few details. For each e , we can view the e^{th} partial computable function φ_e as the function that tells us what the construction is: for each s , we interpret $\varphi_e(s)$ as the code of the state of the construction at stage s . For each e we can calculate a $\Pi_2(A)$ index for the set of α for which $\psi(\alpha)$ holds according to the construction described by φ_e , and using this, a computable index for all the resulting sequences $\bar{\ell}(\psi(\alpha), n)$ and $\bar{\ell}(\neg\psi(\alpha), n)$; and given that, an index $f(e)$ such that $\varphi_{f(e)}$ describes the construction which is performed using these sequences. A fixed point $\varphi_e = \varphi_{f(e)}$ is the construction that we actually perform. Note that regardless if φ_e is partial or not, $\varphi_{f(e)}$ is always total, and so the fixed point is total.

Lemma 2.4. For all α :

- (a) For all s , $\ell(\alpha)_s \leq s$;
- (b) If $\alpha \leq \beta$ then $\ell(\alpha)_s \geq \ell(\beta)_s$.
- (c) If $\bar{\ell}(\alpha)$ is unbounded then there is some child β of α such that $\bar{\ell}(\beta)$ is unbounded.

Proof. Mostly immediate; (c) holds because for every α , one of $\psi(\alpha)$ and $\neg\psi(\alpha)$ holds. \square

Also, by definition, $\bar{\ell}(\lambda)$ is unbounded. We can therefore define:

Definition 2.5. The *true path* is the path of nodes α such that $\bar{\ell}(\alpha)$ is unbounded, but for every $\beta <_L \alpha$, $\bar{\ell}(\beta)$ is bounded.

In other words, λ is on the true path, and if α is on the true path, then the child of α on the true path is the leftmost child β for which $\bar{\ell}(\beta)$ is unbounded. This is the leftmost child that guesses a correct witness for the Σ_3 version of $\psi(\alpha)$ or its negation. Lemma 2.4(c) implies that the true path is infinite.

Pulling and eliminating. We will now discuss the heart of the construction: how to decide where balls move, and which are enumerated into M . The following terminology will be useful.

- A **fin-node** is a nonzero node β whose last entry is \mathbf{fin}_n for some n (that is, $\beta = (\beta^-) \hat{\ } \mathbf{fin}_n$ for some n); and similarly,
- An ∞ -**node** is a nonzero node whose last entry is ∞_n for some n .

Here note that for a nonzero node α , we let α^- denote α 's parent, the result of removing the last entry of α .

The next definition will govern the movement and enumeration of balls. We will explain the details after we give the definition. The rough idea, though, is the following. If β is an ∞ -node of length $e + 1$, then we will ensure that $Y(\beta) \subseteq W_e$ by only allowing elements that have already entered W_e to enter $Y(\beta)$. On the other hand, if β is a fin-node, then we get $Y(\beta) \cap W_e =^* \emptyset$ "for free", by the fact that the β^- question $\psi(\beta^-)$ fails. As discussed, in the first case, an ∞ -node β will want to enumerate into M any balls it sees that are outside $W_{e,s}$. If β is correct about its guess, then this will not cause $\mathbb{N} \subseteq^* M$, because $\psi(\beta^-)$ holds: sufficiently many balls will not be enumerated into M . The complexity of the definitions comes from the need to deal with nodes to the left of the true path, that have wrong opinions.

For any node β , let

$$Y(<_L \beta)_s = \bigcup \{Y(\gamma)_s : \gamma <_L \beta\},$$

and let

$$Y(\leq \beta)_s = Y(<_L \beta)_s \cup Y(\beta)_s = \bigcup \{Y(\gamma)_s : \gamma \leq \beta\}.$$

Here recall that $\gamma \leq \beta$ denotes the Kleene-Brouwer ordering, not the lexicographic ordering.

Definition 2.6. A ball x is *pullable* by a nonzero node β at stage s if:

- (i) $x \in Y(\beta^-)_s \setminus Y(\leq \beta)_s$;
- (ii) $x > |\beta|$;
- (iii) $|Y(\beta)_s| < \ell(\beta)_s$; and
- (iv) If β is an ∞ -node then $x \in W_{|\beta|,s}$.

A ball x is *eliminable* by a nonzero node β at stage s if:

- (v) $x \in Y(\beta^-)_s \setminus Y(\leq \beta)_s$;
- (vi) $x \neq \min Y(\beta^-)_s$; and
- (vii) $x < \ell(\beta)_s$.

Let us explain. (i) and (v) mean that x resides at β 's parent β^- , or at or below one of β 's siblings to the right of β . These are the regions from which β is allowed to pull. Thus, balls on the machine will only move downward in the Kleene-Brouwer ordering: either to the left, or drop from a parent to a child.

(ii) will help ensure that each ball outside M eventually stops moving: it cannot keep being pulled by longer and longer nodes. (vi) will help show that M is coinfinite. A node β^- on the true path will guard one ball (the smallest that it can see), and ensure that it is not enumerated into M .

The fact that an ∞ -node β is only allowed to pull balls in W_e is in conflict with the requirement that $Y(\beta) =^* M^c$. It is simple to reconcile the conflict: if at a stage s , x is in a region from which β can pull, and $x \notin W_{e,s}$, then β can simply enumerate x into M and remove it from the machine. The danger, of course, is that numbers go into W_e slowly, and we may be too hasty in enumerating them into M , risking $M =^* \mathbb{N}$. To avoid this, we use the sequences $\ell(\gamma)$, that will ensure that each node γ to the left of the true path eventually stops enumerating into M numbers from A^c . If β is an ∞ -node on the true path, we need to ensure that $Y(\beta)$ is infinite, i.e., that it sees enough numbers in W_e sufficiently early. We will show that this follows from $\psi(\beta^-)$ being true.

Thus, (iii) and (vii) ensure that nodes to the left of the true path eventually stop acting: they stop eliminating or pulling balls $x \in A^c$. (Balls $x \in A$ are ‘‘phantom’’ balls; the right way to think about them is as if they were never there, and so pulling them does not count as acting.) This uses the fact that if γ lies to the left of the true path, then $\ell(\gamma)$ is bounded. Of course, these restrictions also restrain the action of a node β on the true path. Here, we note the difference between (iii) and (vii), i.e., the difference between the conditions $x < \ell(\beta)_s$ and $|Y(\beta)_s| < \ell(\beta)_s$. The issue is of timing.

The main step of the verification will be to ensure that if β lies on the true path, then the number of balls in $Y(\beta)$ during A -true stages is unbounded (this is Lemma 2.14 below). Suppose that this is known for β^- (either by induction, or directly from $\psi(\beta^-)$, if β is an ∞ -node). We need to ensure that if the opportunity arises, in a true stage, then β will act by pulling balls. It could be that at such stages, $\ell(\beta)_s$ is too small, compared to the size of the balls that need pulling. This is why we do not require that $x < \ell(\beta)_s$ when deciding whether to pull x or not; we need to consider the number of balls we currently have, i.e., $|Y(\beta)_s|$.

On the other hand, there is no reason for haste in eliminating balls; we just need to ensure that in the limit, all but finitely many ‘‘deviant’’ balls do end up in M (these are the balls outside W_e , when β ensures that $M^c \subseteq^* W_e$). So it is legitimate for us to require the stronger condition $x < \ell(\beta)_s$ when considering elimination. And in fact, it is important that we do so. The reason is a little delicate. Suppose that α , the node on the true path, is a **fin**-node, but that γ is an ∞ -child of α^- that lies to the left of α . Since γ is allowed to only pull elements of W_e (again $e = |\alpha^-|$), and W_e may be small (indeed, may be empty), it may be the case that $Y(\gamma)$ never reaches the size $\lim_s \ell(\gamma)_s$. Thus, γ is always ‘‘hungry’’ for balls. If we allowed γ to eliminate balls whenever $|Y(\gamma)_s| < \ell(\gamma)_s$, it may never stop doing so. For pulling

balls, the weaker condition $|Y(\gamma)_s| < \ell(\gamma)_s$ is sufficient, since if γ keeps pulling balls $x \notin A^c$, we will argue that eventually the size of $Y(\gamma)$ reaches $\lim_s \ell(\gamma)_s$, and γ will stop acting.

We will need the following for the construction:

Lemma 2.7. *If x is a ball on the machine at some stage s , and is pullable by some node β at stage s , then there is a \leq -least such β .*

Proof. There are only finitely many nodes α such that $x \in Y(\alpha)_s$, namely those $\alpha \leq \gamma$ where $x \in S(\gamma)_s$. The ball x is pullable only by children of such nodes α , and ones which are $\leq \gamma$. The order-type of all of these nodes (according to the Kleene-Brouwer ordering) is a well-ordering, namely of order-type ω (finitely many to the left of γ , and the children of γ). \square

Construction. At the beginning of stage s , we already have specified M_s and $Y(\alpha)_s$ for all α . We start with $M_0 = \emptyset$, and no ball is on the machine at the beginning of stage 0.

At stage s there are three possibilities.

- (a) If there is a ball on the machine which is pullable by some node, then for each such ball x , let β be the \leq -least node by which x is pullable; we move x to β (by setting $x \in S(\beta)_{s+1}$).⁹
- (b) If no ball on the machine is pullable by some node, but some ball on the machine is eliminable by some node, then we enumerate each such x into M_{s+1} and remove it from the machine.
- (c) If no ball on the machine is either pullable or eliminable, then s is declared to be a new balls stage. Recall that this means that we set $n(s+1) = n(s) + 1$, i.e., that possibly $A_{s+1} \neq A_s$.
 - (i) We let $M_{s+1} = M_s \cup A_{s+1}$, and remove any $x \in A_{s+1}$ from the machine.
 - (ii) We place any $x \in Q_{s+1} \setminus M_{s+1}$ which is not already on the machine, at the root of the machine, i.e., we put it into $S(\lambda)_{s+1}$.

Verification.

Lemma 2.8. *For all s , $Y(\lambda)_s = Q_s \setminus M_s$.*

Proof. By induction on s . If $x \in Y(\lambda)_{s+1} \setminus Y(\lambda)_s$ then $x \in Q_{s+1} \setminus M_{s+1}$; if $x \in Q_s \setminus Q_{s+1}$ then $x \in A_{s+1}$ and so $x \in M_{s+1}$ and so $x \notin Y(\lambda)_{s+1}$. \square

Finite ball movement. As mentioned above, we say that x is a *permanent resident* of a node β if $x \in S(\beta)_s$ for all but finitely many stages s . We let $S(\beta)$ be the set of permanent residents of β .

Lemma 2.9. *If $x \in \bigcup_s Y(\lambda)_s$ and $x \notin M$ then x is a permanent resident of some node.*

Proof. Since $x \notin M$, for all but finitely many s , we have $x \in Y(\lambda)_s$. By (ii) of Definition 2.6, if x resides at some node β at some stage, then $|\beta| \leq x$. Among such nodes, x only moves downward in the Kleene-Brouwer ordering. Since the subtree of nodes of length $\leq x$ is well-founded (has no infinite path), the Kleene-Brouwer

⁹Observe that in this case, we move *all* balls on the machine that are pullable by some node, not only by the “strongest” or “highest priority” node, a notion that we did not define.

ordering restricted to such nodes is well-founded, and so, x cannot move infinitely often. \square

Lemma 2.9 implies:

Lemma 2.10. *There are infinitely many new balls stages.*

Thus, as $n(s) \rightarrow \infty$, every $x \notin M$ is eventually put on the machine and is never removed from the machine. This shows:

Lemma 2.11. *Every $x \notin M$ is a permanent resident of some node.*

We let $Y(\alpha) = \lim_s Y(\alpha)_s = \bigcup_{\beta \succ \alpha} S(\beta)$; so $Y(\lambda) = M^c$.

True path. Recall that the true path (Definition 2.5) is the collection of nodes α such that $\bar{\ell}(\alpha)$ is unbounded, but for all $\gamma <_L \alpha$, $\bar{\ell}(\gamma)$ is bounded. We observed that the true path is an infinite path of the tree.

Lemma 2.12. *If α lies on the true path, then there are only finitely many stages at which some node $\gamma <_L \alpha$ either pulls or eliminates a ball $x \in A^c$.*

Proof. This is proved by induction on the length of α ; it is vacuously true for $\alpha = \lambda$. Suppose that this is known for α^- . Let

$$l = \max \left\{ \lim_{s \rightarrow \infty} \ell(\beta)_s : \beta <_L \alpha \text{ is a child of } \alpha^- \right\}.$$

By Lemma 2.4(b),

$$l = \max \left\{ \lim_{s \rightarrow \infty} \ell(\gamma)_s : \gamma <_L \alpha \text{ \& } \gamma > \alpha^- \right\}.$$

Then by (vii) of Definition 2.6, no number $x \geq l$ is ever eliminated by any $\gamma <_L \alpha$ extending α^- (indeed, whether it is an element of A^c or not). By induction (from left to right) on the children β of α^- which lie to the left of α , we show that β pulls only finitely many $x \in A^c$. Suppose that this is true for all $\gamma <_L \beta$. Then after some stage, every ball $x \notin A$ which is pulled by β , is not later pulled by a node to the left of β , and is not eliminated by any descendant of β , and so is an element of $Y(\beta)$. So if β pulls infinitely many $x \in A^c$, then $Y(\beta)$ is infinite. But then, $|Y(\beta)_s| \geq l$ for all but finitely many stages s . By (iii) of Definition 2.6, at such stages, β pulls no balls, a contradiction. \square

As a result:

Lemma 2.13. *If α lies on the true path, then $Y(<_L \alpha)$ is finite.*

The main lemma is the following.

Lemma 2.14. *If α lies on the true path, then for every k there is some A -true stage s such that $|Y(\alpha)_s| \geq k$.*

Proof. We prove this by induction on the length of α . First we consider $\alpha = \lambda$. If s is a new balls stage and $s+1$ is A -true, then, as discussed above, at least $n(s)$ many new balls from Q_{s+1} are placed in $S(\lambda)_{s+1}$; and $n(s) \rightarrow \infty$.

Suppose that $\alpha \neq \lambda$ lies on the true path, and that the lemma holds for α^- . Let s_0 be a stage after which no $\gamma <_L \alpha$ pulls or eliminates any ball $x \in A^c$; in particular, no $\gamma <_L \alpha$ pulls any ball during an A -true stage $s > s_0$. Further, if $s > s_0$ is A -true then $Y(<_L \alpha)_s = Y(<_L \alpha)$, as all balls on the machine at stage s are from A^c . Let $m = |Y(<_L \alpha)|$, which by Lemma 2.13 is finite.

Let $k \in \mathbb{N}$. There are two cases.

First, suppose that α is a **fin**-node. By induction, there is an A -true stage $s > s_0$ for which

$$|Y(\alpha^-)_s| \geq k + m + 1.$$

Choose such s sufficiently large so that $\ell(\alpha)_s > k$. If $|Y(\alpha)_s| \geq k$ then we are done. Otherwise, $|Y(\alpha)_s| < \ell(\alpha)_s$. Every $x \in Y(\alpha^-)_s \setminus Y(\leq \alpha)_s$ larger than $|\alpha|$ is pullable by α at stage s . Now $Y(\alpha^-)_s$ contains at most one $x \leq |\alpha|$, namely $x = |\alpha|$ (as every $x \in Y(\alpha^-)_s$ has size $> |\alpha^-|$)¹⁰; and as discussed, $|Y(<_L \alpha)_s| = m$. Hence, there remain at least $k - |Y(\alpha)_s|$ many balls in $Y(\alpha^-)_s$ that are pullable by α . Since $s > s_0$, and s is A -true, no node to the left of α pulls such balls at stage s . Hence, at stage s , all balls pullable by α will actually be pulled by α , and moved to $Y(\alpha)_{s+1}$. Also, no balls already in $Y(\alpha)_s$ will be pulled to the left. Since balls are pulled at stage s , no balls are eliminated (anywhere on the machine) at stage s — we only eliminate balls if no balls are pullable. So $Y(\alpha)_s \subseteq Y(\alpha)_{s+1}$, and overall, we see that $|Y(\alpha)_{s+1}| \geq k$. Since s is not a new balls stage, $s + 1$ is also A -true, and so is as required.

If α is an ∞ -node, then the argument is the same, except that we need an A -true stage $s > s_0$ such that $\ell(\alpha)_s > k$ and such that

$$|Y(\alpha^-)_s \cap W_{|\alpha^-|,s}| \geq k + m + 1,$$

since α is allowed to pull only balls from $W_{|\alpha^-|,s}$. However, since $\bar{\ell}(\alpha)$ is unbounded, $\psi(\alpha^-)$ holds, which gives us exactly what we need. Note that in this case, we do not need to use the inductive hypothesis on α^- , nor is it sufficient for our purposes.¹¹ \square

Remark 2.15. During the proof of Lemma 2.14, we relied on our stipulation that we do not eliminate any balls during a stage at which some balls are pullable. So we showed that we can get $|Y(\alpha)_{s+1}| \geq k$, but it is possible that immediately after that, some balls from $Y(\alpha)_{s+1}$ are eliminated, reducing the size of the set. It seems quite silly that Lemma 2.14 would depend on such an unimportant timing trick.

Indeed, it does not. We could allow balls to be pulled and eliminated at the same stage. But then, in order to prove Lemma 2.14, we would need to first show that if more and more balls are supplied to $Y(\alpha)$, then many of these balls would not be eliminated. We will now do this; it is merely a convenience for our presentation, to delay this part of the argument, instead of essentially proving Lemma 2.14 and Lemma 2.16 together.

M is not everything.

Lemma 2.16. *For every α which lies on the true path, $Y(\alpha) \neq \emptyset$.*

Proof. Let α be a node which lies on the true path; let s_0 be a stage after which no ball $x \notin A$ is pulled or eliminated by any node $\gamma <_L \alpha$. Let x be the smallest number in A^c for which there are stages $t \geq s > s_0$ such that:

- $x \in Y(\alpha)_t$;
- s is A -true and $x \leq \max Y(\alpha)_s$.

¹⁰For the purposes of this argument, let us agree that $0 \notin \mathbb{N}$, so that this calculation applies to $\alpha^- = \lambda$ as well.

¹¹Curiously, this means that we would not need to prove Lemma 2.14 by induction, if we already knew that there are infinitely many ∞ -nodes on the true path. This is true, however, we need Lemma 2.14 to prove this.

Such x exists by Lemma 2.14 (we can take $t = s$). Let stages $t \geq s > s_0$ witness x .

By induction on stages $r \geq t$, we argue that $x \in Y(\alpha)_r$ and that $x = \min Y(\alpha)_r$. The minimality of x ensures that $x = \min Y(\alpha)_t$. Let $r \geq t$, and suppose that $x = \min Y(\alpha)_r$. By Definition 2.6(vi), at stage r , x is not eliminable by any child of α ; since $r > s_0$, x is not eliminated by any node at stage r . Similarly, since $r > s_0$, x is not pulled by any node to the left of α at stage r . Hence, $x \in Y(\alpha)_{r+1}$. Any $y < x$ on the machine at stage $r + 1$ is an element of A^c , as s was A -true, and so by minimality of x , will not enter $Y(\alpha)_{r+1}$. \square

As a result:

Lemma 2.17. *M is coinfinite.*

Proof. Since the true path is infinite, for every finite $D \subseteq M^c$ there is some α on the true path such that $D \cap Y(\alpha) = \emptyset$; $Y(\alpha) \subseteq M^c$ and so by Lemma 2.16, $M^c \neq D$. \square

Maximality.

Lemma 2.18. *For every α on the true path, $Y(\alpha) =^* M^c$.*

Proof. We know this holds for $\alpha = \lambda$, and so it suffices to show that if $\alpha \neq \lambda$ lies on the true path then $Y(\alpha) =^* Y(\alpha^-)$. By Lemma 2.13, $Y(<_L \alpha)$ is finite, so it suffices to show that all but finitely many elements of $Y(\alpha^-) \setminus Y(<_L \alpha)$ are in $Y(\alpha)$. Let $x \in Y(\alpha^-) \setminus Y(<_L \alpha)$, and suppose that $x > \min Y(\alpha^-)$.

If s is sufficiently late, then $\ell(\alpha)_s > x$, $x \in Y(\alpha^-)_s \setminus Y(<_L \alpha)_s$, and $x > \min Y(\alpha^-)_s$. Then x is eliminable by α at s . Hence, x must in fact be pulled by α at stage s . \square

Remark 2.19. Recall that the reason for eliminating balls is when we want $M^c \subseteq^* W_e$; we then need to enumerate the complement of W_e into M . However, this was not incorporated into the definition of “eliminable”, and it seems that Lemma 2.18 relies on elements of W_e being eliminable by **fin**-nodes (when $|Y(\alpha)_s| \geq \ell(\alpha)_s$). This is not necessary. In Definition 2.6, we could add the clause “ β is an ∞ node and $x \notin W_{|\alpha^-|,s}$ ” to the definition of eliminability.¹² But then we would need to separate the proof of Lemma 2.18 into cases. If α is an ∞ -node, then the proof is as above. If it is a **fin**-node, then we argue that as $\bar{\ell}(\alpha)$ is unbounded, it will eventually pull balls x as above.

The proof of Theorem 2.1 is concluded with the following lemma.

Lemma 2.20. *M is maximal.*

Proof. Let $e \in \mathbb{N}$; let α be the node on the true path of length $e + 1$.

First, suppose that α is an ∞ -node. Every ball pulled by α is already an element of W_e , so $Y(\alpha) \subseteq W_e$. By Lemma 2.18, $M^c \subseteq^* W_e$.

Next, suppose that α is a **fin**-node. Since $\bar{\ell}(\alpha)$ is unbounded, we know that $\psi(\alpha^-)$ fails: there is some k such that for every A -true stage s ,

$$|Y(\alpha^-)_s \cap W_{e,s}| \leq k.$$

¹²This would make it easier for us to allow pulling and eliminating balls at the same stage, if we so wished, as it would ensure that no ball is both pullable and eliminable by the same node; see Remark 2.15.

Then $|Y(\alpha^-) \cap W_e| \leq k$. For otherwise, there would be a set $D \subseteq Y(\alpha^-) \cap W_e$ of size $k + 1$. But then, for all but finitely many stages s , $D \subseteq Y(\alpha^-)_s \cap W_{e,s}$, so $|Y(\alpha^-)_s \cap W_{e,s}| \geq k + 1$ for some A -true stage s , which is not the case. It follows that $M^c \cap W_e =^* \emptyset$. \square

3. ATOMLESS SUPERSETS

We modify the construction above to prove Theorem 1.2: every coinfinite low₂ c.e. set has an atomless, hyperhypersimple superset.

The new ingredient is a process for splitting a stream of balls in two, each infinite outside of A .

Boolean algebras and binary trees. Recall that we can generate Boolean algebras from trees. If $T \subseteq 2^{<\omega}$ is a tree, then $B(T)$ is the quotient of the free Boolean algebra with generators $\sigma \in T$, modulo the relations:

- Every $\tau \in T$ is the join of its children;
- If $\sigma, \tau \in T$ are incomparable then $\sigma \wedge \tau = 0_{B(T)}$.

Equivalently, we can think of $B(T)$ as the collection of finite unions of the sets $[\sigma] \cap [T]$, for all $\sigma \in T$, ordered by set inclusion; in this version, τ is identified with $[\tau] \cap [T]$.

Note that this implies that if τ is a leaf (or more generally, if τ is not extendible to an infinite path on T), then $\tau = 0_{B(T)}$; and that $\lambda = 1_{B(T)}$. Every countable Boolean algebra can be presented in such a way (up to isomorphism) by considering a tree of nonzero finite Boolean combinations of any sequence of generators.¹³ For our purposes this section, we rely on the fact that the atomless Boolean algebra is $B(2^{<\omega})$, the one generated by the full binary tree.¹⁴ Thus, given a coinfinite low₂ c.e. set A , we will enumerate a superset $H \supseteq A$, and construct a tree of sets $\langle Z(\rho) : \rho \in 2^{<\omega} \rangle$ satisfying:

- (i) $Z(\lambda) =^* H^c$;
- (ii) For each ρ , $Z(\rho)$ is infinite;
- (iii) For each ρ , $H \cup Z(\rho)$ is c.e.;
- (iv) For all $\rho \in 2^{<\omega}$, $Z(\rho \hat{0})$ and $Z(\rho \hat{1})$ are disjoint, and $Z(\rho) =^* Z(\rho \hat{0}) \cup Z(\rho \hat{1})$;
- (v) For every c.e. set $W \supseteq H$ there is a finite set $D \subset 2^{<\omega}$ such that $W \setminus H =^* \bigcup \{Z(\rho) : \rho \in D\}$.

This suffices: to see that $\mathcal{L}^*(H)$ is a Boolean algebra, it suffices to show that it is complemented (see [Soa87, Sec. X.2]). Indeed, if $W \supseteq H$ is c.e., by (v), let $D \subseteq 2^{<\omega}$ be a finite set such that $W \setminus H =^* \bigcup \{Z(\rho) : \rho \in D\}$. By (iv), we may assume that all elements of D have the same length, say k . Let $E = \{0, 1\}^k \setminus D$; then $H \cup \bigcup \{Z(\rho) : \rho \in E\}$ is a complement of W in $\mathcal{L}^*(H)$; by (iii), it is c.e. (ii) and (iv) also ensures that $\mathcal{L}^*(H)$ has no atoms.

As we are using the Δ_3 machinery, the sets $Z(\rho) \cup H$ will not be uniformly c.e.; rather, they will be $=^*$ to sets whose indices we can read off the true path of the construction.

¹³In greater detail: if $\{b_n\}$ is a set of generators of B , then we let T be the set of all sequences σ such that $b_\sigma = \bigwedge \{b_i : \sigma(i) = 1\} \wedge \bigwedge \{\neg b_i : \sigma(i) = 0\}$ is nonzero. See the proof of [Soa87, Thm. X.7.2]

¹⁴In general, the atoms of $B(T)$ are those nodes that isolate a path, so if $T \subseteq 2^{<\omega}$ has no leaves, then $B(T)$ is atomless if and only if it is perfect. Among all perfect trees, we choose $T = 2^{<\omega}$ as it makes notation simplest.

The setup. We will use the same general mechanism as above. We will have an infinite-branching tree of strategies, that will serve as a pinball machine on which we move balls around. When numbers are enumerated into H , they will be removed from the machine; the complement H^c will consist of the balls that have permanent residence at some node on the tree.

Toward enumerating the sets $Z(\rho)$, each ball x on the machine at some stage s will have a label $\rho \in 2^{<\omega}$. As with residence on the tree, the label of a ball $x \notin H$ will stabilise (indeed, the label can change only when x moves). If the final label of x is ρ , then we will put x into $Z(\rho')$ for all $\rho' \leq \rho$.

We will use the notation $S(\alpha)_s, Y(\alpha)_s, Y(<_L \alpha)_s$ etc. as above. We will use similar notation that also specifies labels. Namely, for a node α on the tree and $\rho \in 2^{<\omega}$, we will let $S(\alpha, \rho)_s$ denote the collection of all balls x which at the beginning of stage s , reside at α and have label ρ . We will let

$$Y(\alpha, \rho)_s = \bigcup \{S(\alpha', \rho')_s : \alpha \leq \alpha' \ \& \ \rho \leq \rho'\}.$$

That is, $Y(\alpha, \rho)_s$ consists of the balls x which at the beginning of stage s ,

- reside at some $\alpha' \geq \alpha$, and
- have a label $\rho' \geq \rho$.

So as discussed, the elements of $Y(\alpha, \rho)_s$ are those elements in $Y(\alpha)_s$ that at stage s we intend to put in $Z(\rho)$. So at the end of the verification, we will let $Z(\rho) = Y(\lambda, \rho)$, the collection of all balls $x \in Y(\lambda)$ whose permanent label extends ρ . As in the previous construction, if α is on the true path, then $Y(\alpha) =^* H^c$, so $Z(\rho) =^* Y(\alpha, \rho)$.

The requirements. In the current construction, we will have two kinds of requirements. The analogues of the maximality requirements from the previous construction are the requirements for meeting (v) above. For each $\rho \in 2^{<\omega}$ of length e , we will ensure that either $Z(\rho) \subseteq^* W_e$ or $Z(\rho) \cap W_e =^* \emptyset$. Thus, $D = \{\rho \in \{0, 1\}^e : Z(\rho) \subseteq^* W_e\}$ will show that (v) holds for W_e . The way to meet these requirements will be very similar to the previous construction, except that the α question will be more complicated; for each $\rho \in \{0, 1\}^e$, we will need to guess whether we will see enough balls to make $Z(\rho) \subseteq W_e$ while ensuring that $Z(\rho)$ is infinite. The mechanism of pulling and eliminating will be used to achieve that, but we will see that the definitions of “pullable” and “eliminable” need to be more complicated.

The other requirements are new: they ensure that (iv) above holds. Namely, for each ρ , we need to ensure that $Z(\rho \hat{\ } 0)$ and $Z(\rho \hat{\ } 1)$ form a splitting of $Z(\rho)$ (up to finite differences), into two infinite sets. In the construction, we will have nodes devoted to such a requirement. The task for such a node is to take balls with label ρ , and decide whether to extend their label to either $\rho \hat{\ } 0$ or $\rho \hat{\ } 1$. The difficulty, of course, is that during the construction, we do not know whether a ball x is in A or not. It would be very bad if all but finitely many balls x that we direct to $\rho \hat{\ } 0$, for example, will end up in A , as that would make $Z(\rho \hat{\ } 0)$ finite. Unlike the other kind of requirement, Δ_3 guessing is not sufficient for this task: we know that there will be infinitely many $x \notin A$ with label ρ . We need to somehow obtain individual balls that we have a good reason to guess are not in A .

Our main contribution is precisely this: a new method for *certifying* that groups of balls are not in A . As we will explain later, in fact, to perform this certifying and splitting, it will be notationally convenient to have not just one level of nodes devoted

for each requirement, but two; nodes and their children together will perform these tasks.

So we will have two kinds of nodes.

- (i) *Decision nodes*, of length $3e$, whose task, for each $\rho \in \{0, 1\}^e$, to decide W_e on $Y(\alpha, \rho)$. We call a node of length $3e$ an *e-decision node*.
- (ii) *Splitting nodes* and their *children*, of lengths $3e + 1$ and $3e + 2$. The children of a decision node α of length $3e + 1$ will pull balls with labels ρ of length e , and decide to extend their label to either $\rho\hat{0}$ or $\rho\hat{1}$. We call a node of length $3e + 1$ a *parent e-splitting node*, and a node of length $3e + 2$ a *child e-splitting node*.

True balls rather than true stages. As above, we will make use of a true-stage enumeration (\tilde{A}_r) of A with respect to the function $f(r) = r^2$. We will use the same mechanism as above to slow down this enumeration to an enumeration (A_s) defined during the construction, again letting $n(s) = r$ when $A_s = \tilde{A}_r$. As in the previous construction, if we move or enumerate balls, or perform any other action, at a stage s , then will set $A_{s+1} = A_s$ (by setting $n(s + 1) = n(s)$). When no other action is taken at stage s , we will declare s to be a new balls stage, and set $n(s + 1) = n(s) + 1$.

As in the very first part of the proof of Lemma 2.14, this ensures that infinitely often, the *root* of the tree receives large collections of balls which are all outside A .

However, unlike the previous construction, there will be an element of delay to ball movement. In order to ensure that both $Y(\alpha, \rho\hat{0})$ and $Y(\alpha, \rho\hat{1})$ receive many balls outside A , a child of a splitting node α will hold on to many balls in $Y(\alpha, \rho)$ until it receives confirmation from \mathcal{O}' that these balls are indeed outside A . (Infinitely often, this confirmation will be incorrect, but infinitely often it will be correct, and this will be sufficient for our purposes; we will see that what is important, is that each *individual attempt* at certification makes only finitely many mistakes.) The stages at which such confirmation is received do not need to line up with the A -true stages. So we will not be able to directly prove Lemma 2.14 for the current construction. Rather, instead of looking for “completely true” *stages*, at which every ball on the machine is correct, we will simply ensure that we get more and more true *balls*.

Definition 3.1. A number $x \in A_s^c$ is *A-true* at stage s if

$$A \upharpoonright x + 1 = A_s \upharpoonright x + 1.$$

That is, not only x will not enter A in the future, but no number $y \leq x$ currently outside A will enter A in the future. If we get enough of these, we do not really care that larger balls are not A -true at the same stage.

Definition 3.2. As in the previous construction, we let Q_s be the set of the $n(s)^2$ -smallest elements of A_s^c . We let C_s denote the collection of $x \in Q_s$ which are A -true at stage s .

The questions and the tree. A decision node α needs to decide, for each ρ of length e , whether to make $Z(\rho)$ almost contained in W_e or almost disjoint from W_e . This means that the α question is more complicated. Let α be an e -decision node. For each set $D \subseteq \{0, 1\}^e$, the statement $\psi(\alpha, D)$ says:

D is the set of $\rho \in \{0, 1\}^e$ for which for every k there is some s such that

$$|Y(\alpha, \rho)_s \cap W_{e,s} \cap C_s| \geq k.$$

The outcomes of α are

$$D_n$$

for each $D \subseteq \{0, 1\}^e$ and each $n \in \mathbb{N}$, ordered in order-type ω in some way. Note that $\psi(\alpha, D)$ will hold for precisely one $D \subseteq \{0, 1\}^e$, and we will ensure that $Z(\rho) \subseteq^* W_e$ if and only if $\rho \in D$ (and otherwise, $Z(\rho) \cap W_e =^* \emptyset$).

Since each $\psi(\alpha, D)$ is a finite Boolean combination of $\Pi_2(A)$ statements, it is equivalent to a Σ_3 statement, so as in the previous construction, we obtain uniformly computable, nondecreasing sequences $\bar{\ell}(\psi(\alpha, D), n)$, such that $\psi(\alpha, D)$ holds if and only if for some n , $\bar{\ell}(\psi(\alpha, D), n)$ is unbounded. As above, we use these to define sequences $\bar{\ell}(\alpha)$ for nodes α ; these will only be used for nodes α that are the *children* of decision nodes. So we recursively define:

- If $\alpha = (\alpha^-) \hat{\ } D_n$ is the child of a decision node α^- , then we let $\ell(\alpha)_s$ be the minimum of $\ell(\psi(\alpha^-, D), n)_s$, and $\ell(\beta)_s$ for any $\beta \preceq \alpha$ that is also the child of a decision node.

It is more difficult to explain now why we need both parent and children splitting nodes. If α is a parent e -splitting node, then its children are $\alpha \hat{\ } n$ for all $n \in \mathbb{N}$, ordered naturally. If β is such a child, then β has a unique child β^+ on the tree (which will in turn be an $e + 1$ -decision node). The outcomes n of α do not quite represent guesses about the behaviour of α ; we will discuss them later.

Updating previous definitions. We update notation from the previous construction. For a node α and $\rho \in 2^{<\omega}$, and a stage s , we let:

- $Y(<_L \alpha, \rho)_s$ be the collection of balls that at the beginning of stage s , reside at some node β that lies to the left of α , and have label extending ρ . That is, $Y(<_L \alpha, \rho)_s = \bigcup_{\beta <_L \alpha} Y(\beta, \rho)_s = Y(<_L \alpha)_s \cap Y(\lambda, \rho)_s$;
- $Y(\leq \alpha, \rho)_s = Y(<_L \alpha, \rho)_s \cup Y(\alpha, \rho)_s$.

Definition 3.3. Let $\alpha = (\alpha^-) \hat{\ } D_n$ be a child of an e -decision node, and let $\rho \in \{0, 1\}^e$.

A ball x is *pullable* by (α, ρ) at stage s if the following all hold:

- (i) $x \in Y(\alpha^-, \rho)_s \setminus Y(\leq \alpha, \rho)_s$;
- (ii) $x > |\alpha|$;
- (iii) either:
 - $|Y(\alpha, \rho)_s| < \ell(\alpha)_s$, or
 - $Y(\alpha, \rho)_s \neq \emptyset$ and $x < \max Y(\alpha, \rho)_s$;
- (iv) if $\rho \in D$ then $x \in W_{e,s}$.

A ball x is *eliminable* by (α, ρ) at stage s if:

- (v) $x \in Y(\alpha^-, \rho)_s \setminus Y(\leq \alpha, \rho)_s$;
- (vi) $x \neq \min Y(\alpha^-, \rho)_s$; and
- (vii) $x < \ell(\alpha)_s$.

We need to comment on the new part of this definition, namely the second part of (iii). The issue is that, as mentioned above, we will not be able to directly show that $Y(\alpha)_s$ is “full” during A -true stages (in the sense of Lemma 2.14), only that it gets “filled” by A -true balls (see Proposition 3.13(b) below). Consider a node α on the true path (a child of a decision node). At a stage s , the parent α^- may have many A -true balls. At the same stage, α already has larger balls and so does not feel “hungry” ($|Y(\alpha, \rho)_s| \geq \ell(\alpha)_s$). But because s is not an A -true stage, the

balls that α has may be false. The problem is solved by allowing α to pull balls smaller than ones it already has, even if it does not feel “hungry”. This modification is sufficiently tame so that nodes on the left of the true path will again only pull finitely many balls from A^c .¹⁵

Certification. Since A is low₂ and c.e., some \emptyset' -computable function dominates all A -computable functions (this follows from Martin’s characterisation [Mar66] of the high degrees as those that compute functions that dominate all computable functions).

Definition 3.4. We fix φ to be a \emptyset' -computable function that dominates all A -computable functions. We also fix a computable approximation $(\varphi_s) = (\varphi_0, \varphi_1, \dots)$ of φ .

Let α be a parent e -splitting node, and fix $\rho \in \{0, 1\}^e$. The very rough idea of splitting is that α will define *blocks* of balls, and hold on to these balls until φ gives confirmation to release these balls to nodes below. The k^{th} block will contain at least $2k$ many balls. For now, let us forget about α ’s children, and imagine that released balls are passed to the next splitting node (if α did not have multiple children then there would be a unique $(e + 1)$ -splitting node extending α). When the k^{th} block of balls is *certified* by φ , then k of the balls in that block will receive the label $\rho \hat{0}$, and the rest, the label $\rho \hat{1}$.

For this purpose, we will define an A -computable function $f^{\alpha, \rho}$. An input $k \in \mathbb{N}$ for $f^{\alpha, \rho}$ indicates an attempt to capture and test the k^{th} block of balls. Suppose that at some stage r , for all $m < k$, the m^{th} -block of balls is currently defined, and that α holds $2k$ many balls with label ρ (that are not associated with any existing block). We would then declare these balls to constitute the k^{th} block, and define $f_{r+1}^{\alpha, \rho}(k) > \varphi_r(k)$. The A_{r+1} -use $u = u_{r+1}^{\alpha, \rho}(k)$ of this computation will bound at least $2k$ -many elements of the block.

We then wait for a stage $s > r$ at which one of two things happens. Either $A_s \upharpoonright u \neq A_{r+1} \upharpoonright u$, in which case $f_s^{\alpha, \rho}(k) \uparrow$. This A -change likely involves some elements of the block entering A . We will then wait for α to obtain new balls, that will allow us to redefine a new k^{th} block.

Otherwise, we hope to see $\varphi_s(k) > f_s^{\alpha, \rho}(k) = f_{r+1}^{\alpha, \rho}(k)$. This means that $\varphi_s(k) > \varphi_r(k)$. We will regard this as evidence that elements from the block will not later enter A , and say that the block is *certified*. We then process the block as described above, splitting it evenly between $\rho \hat{0}$ and $\rho \hat{1}$, and passing the balls in the block to the next node below.

Of course, it is possible that neither of these events happen: the computation $f_{r+1}^{\alpha, \rho}(k)$ is never undefined, but $\varphi(k)$ never exceeds the value of that computation. Therefore, while we wait, we will try to define the $(k + 1)^{\text{th}}$ -block, and so on.

The argument that this works will be by contradiction. We want to argue that infinitely many blocks will be released, and infinitely many of those will be A -correct, i.e., only contain balls from A^c . Assuming this is not the case, we will want to show that $f^{\alpha, \rho}$ is a total function. If this is shown, then we obtain the desired

¹⁵It may be difficult to envision how this situation can come about, since smaller balls are put on the machine earlier than larger balls. However, the collection of balls pullable by a node is not always an initial segment of the collection of available balls, since only some may be elements of some W_e , whereas the others may not yet be eliminable by the same node. So the order “breaks” as balls move on the machine.

contradiction from the fact that φ dominates $f^{\alpha,\rho}$, showing that for almost all k , the correct k^{th} block will in fact be certified and released. To show that $f^{\alpha,\rho}$ is total (under the assumption for contradiction), we use the fact that $\varphi_s(k)$ changes only finitely many times. This implies that the k^{th} block will be released only finitely many times, and so eventually, an A -correct k^{th} block will be defined, showing that $f^{\alpha,\rho}(k) \downarrow$.

Thus, the use of the certification process is in the delay it imposes on the release of balls. If we didn't wait for certification, then it is possible that we would keep redefining and releasing some k^{th} block, none of whose versions is A -correct. It would then be possible that of the blocks that we release, all but finitely many of the balls that we target to $\rho^{\wedge}0$, say, end up in A , resulting in $Z(\rho^{\wedge}1) =^* Z(\rho)$, so the splitting requirement is not met.

We now come to the need for multiple children. The reason is delicate. We want to argue that an A -correct definition of some $f^{\alpha,\rho}(k)$ will eventually be made. Imagine the following sequence of events:

- (1) $f^{\alpha,\rho}(k)$ is defined at some stage.
- (2) While waiting for certification, we also define $f^{\alpha,\rho}(k+1)$ (after all, we do not know where φ starts dominating).
- (3) The $(k+1)^{\text{th}}$ block is certified and released, but not the k^{th} one.
- (4) Some x in the k^{th} block enters A , making both $f^{\alpha,\rho}(k)$ and $f^{\alpha,\rho}(k+1)$ undefined.

Now, all we know, by induction, is that an analogue of Lemma 2.14 holds for α : for all m , there is some stage s at which $C_s \cap Y(\alpha, \rho)_s$ has size at least m . If s is such a stage after these events unfolded, then it is possible that the bulk of $C_s \cap Y(\alpha, \rho)_s$ is actually lying well below α — it is part of the $(k+1)$ -block that was released at step (3) above. Since these balls are not currently residing at α , we cannot use them to define a new k^{th} -block. Unless... we pull them back up to α from wherever they are currently residing.

Now, in all similar constructions, pulling balls from below is a source of innumerable problems. Just as a simple example, it becomes much more difficult to argue that balls outside H will eventually reach a permanent residence, since pulling balls back up is *increasing* in the Kleene-Brouwer ordering. More seriously, it will be difficult to show that the sets $H \cup Y(\gamma, \rho)$, for γ on the true path, are c.e. So we will not do this. A work-around is to allow α to have infinitely many children $\alpha^{\wedge}n$. While waiting for certification, each block will “reside” at one of the children. Indeed the block waiting at a child β will simply be $S(\beta, \rho)_s$.

While the k^{th} block is waiting at $\alpha^{\wedge}n$, all blocks defined later will reside at children of α that lie to the right of $\alpha^{\wedge}n$, namely, children $\alpha^{\wedge}m$ for various $m > n$. When the k^{th} block is dissolved (at step (4) above), it is now fine for $\alpha^{\wedge}n$ to pull all balls from the right, even those that were released to lower nodes, and use them to constitute a new k^{th} block.

One can think of $\alpha^{\wedge}n$ as representing the guess that φ starts majorising $f^{\alpha,\rho}$ from input n . This is not exactly right but the intuition is not far off.

Another point to make is that for distinct ρ , the function φ may start dominating the various functions $f^{\alpha,\rho}$ at different locations. The argument above will show that for each $\rho \in \{0, 1\}^e$, there is some child β that releases ρ -blocks infinitely often. However, to obtain a child of α on the true path, we need the *same* β to work for all ρ . Thus, in the definition of certification below, we will let a child β hold on

to its blocks, until all of them (one for each ρ) are certified, and only then will it release them all.

We can now give the details. Let α be a parent e -splitting node; let $\rho \in \{0, 1\}^e$. As discussed, We will define a function $f_s^{\alpha, \rho}$, with intended oracle A . So at various stages s , for various k , we may have $f_s^{\alpha, \rho}(k) \downarrow$ or not, and if it is defined, then we will declare a use $u = u_s^{\alpha, \rho}(k)$ for this computation. The usual rule applies: if $f_s^{\alpha, \rho}(k) \downarrow$, with use u , and $A_s \upharpoonright u = A_{s+1} \upharpoonright u$, then $f_{s+1}^{\alpha, \rho}(k) \downarrow = f_s^{\alpha, \rho}(k)$ with the same use. If, on the other hand, $A_s \upharpoonright u \neq A_{s+1} \upharpoonright u$, then we declare that $f_{s+1}^{\alpha, \rho}(k) \uparrow$. Note that this will only happen if s is a new balls stage, and during such stages, we do not define new computations $f_{s+1}^{\alpha, \rho}(k)$.

We will define $f_s^{\alpha, \rho}(k)$ only for $k \geq 1$, since there is no point in dealing with a block of size 0.¹⁶

We will ensure that if $k \geq 1$ and $f_s^{\alpha, \rho}(k+1) \downarrow$ then $f_s^{\alpha, \rho}(k) \downarrow$ as well, that is, the domain of $f_s^{\alpha, \rho}$ is a (finite) initial segment of $\mathbb{N} \setminus \{0\}$. See Lemma 3.15 below.

When we define a computation $f_r^{\alpha, \rho}(k)$ (at some stage $r-1$), then we will declare that the k^{th} block is currently *waiting* at some child β of α ; we will record this by setting $k_s^\rho(\beta) = k$ (note that α is determined by β). This notation implies that at most one ρ -block is waiting at β at a given stage.

If $k_s^\rho(\beta) \downarrow = k$ then $f_s^{\alpha, \rho}(k) \downarrow$. Thus, if $k_s^\rho(\beta) \downarrow = k$ but $f_{s+1}^{\alpha, \rho}(k) \uparrow$ then we declare that $k_{s+1}^\rho(\beta) \uparrow$. That is, if the k^{th} block is waiting at β at (the beginning of) stage s , but is dissolved when passing from stage s to stage $s+1$, then no block is waiting at β at stage $s+1$.

If β *releases* balls at stage s , then we will also set $k_{s+1}^{\alpha, \rho}(\beta) \uparrow$, even though $f_{s+1}^{\alpha, \rho}(k) \downarrow$; at stage $s+1$, the block does not wait at β anymore.

We will ensure that if $k_s^\rho(\beta) \downarrow$, then for every child γ of α that lies to the left of β , we have $k_s^\rho(\gamma) \downarrow$ as well (and $k_s^\rho(\gamma) < k_s^\rho(\beta)$). Again, see Lemma 3.15.

Definition 3.5. Let α be a parent e -splitting node, let $\rho \in \{0, 1\}^e$, and let β be a child of α . Let s be a stage.

- (a) A ball x is *pullable* by (β, ρ) at stage s if:
 - (i) $x \in Y(\alpha, \rho)_s \setminus Y(\leq \beta, \rho)_s$; and
 - (ii) $k_s^\rho(\beta) \uparrow$ (that is, no ρ -block is waiting at β at stage s).
- (b) Let $k \geq 1$. We say that (β, ρ, k) is *ready for definition* at stage s if:
 - (iii) $f_s^{\alpha, \rho}(k) \uparrow$, and $f_s^{\alpha, \rho}(k-1) \downarrow$;
 - (iv) $k_s^\rho(\beta) \uparrow$; and
 - (v) Letting $v = u_s^{\alpha, \rho}(k-1)$ if $k > 1$, $v = 0$ otherwise, $S(\beta, \rho)_s$ contains at least $2k$ many balls greater than v .
- (c) We say that β is *certified* at stage s , if for all $\rho \in \{0, 1\}^e$:
 - (vi) $k_s^\rho(\beta) \downarrow$; and
 - (vii) $\varphi_s(k) > f_s^{\alpha, \rho}(k)$, where $k = k_s^\rho(\beta)$.

Lemma 3.6. *Suppose that x is pullable by some node at stage s . Then:*

- (a) *For each node β , there is at most one label ρ such that x is pullable by (β, ρ) .*
- (b) *There is a \leq -least node β such that x is pullable by (β, ρ) for some ρ .*

Proof. For (a), suppose that x is pullable by (β, ρ) . Then either β is a child of an e -decision node or a child e -splitting node, $|\rho| = e$, and $x \in Y(\beta^-, \rho)$. So ρ is the initial segment of length e of x 's label at stage s .

¹⁶And it is notationally simpler than requiring the k^{th} block to have size $2k+2$.

(b) is as in Lemma 2.7. \square

Lemma 3.7. *Let α be a parent e -splitting node, and let $\rho \in \{0, 1\}^e$. Let s be a stage. Suppose that no balls are pullable by any node at stage s . Then there is at most one child β of α and one k such that (β, ρ, k) is ready for definition at stage s .*

Proof. The number k is unique by (iii) of Definition 3.5. Say $\beta < \beta'$ are children of α . If $k_s^\rho(\beta) \uparrow$, then every $x \in Y(\beta', \rho)_s$ is pullable by (β, ρ) ; so by assumption on s , $Y(\beta', \rho)_s = \emptyset$, whence (β', ρ, k) cannot be ready for definition at s by (v). \square

Construction. We start with $H_0 = \emptyset$ and no ball on the machine.

At stage s we operate according to the first case that applies.

- (a) There is a ball on the machine which is pullable by some (β, ρ) . For each such ball x , let β be the \leq -least such. We move x to reside at β at stage $s + 1$, and we set x 's label at stage $s + 1$ to be ρ .
- (b) There is a ball x which is eliminable by some pair (β, ρ) . We enumerate all such x into H_{s+1} and remove them from the machine.
- (c) There is some child e -splitting node that is certified at stage s . In this case, for each such β for which there is no $\gamma <_L \beta$ that is also certified at s , for each $\rho \in \{0, 1\}^e$, letting $k = k_s^\rho(\beta)$, we:
 - Move all $x \in S(\beta, \rho)_s$ to the unique child β^+ of β ;
 - For all children $\gamma \geq \beta$ of α , we set $k_{s+1}^\rho(\gamma) \uparrow$;
 - For the k -least many balls $x \in S(\beta, \rho)_s$, we change their label to $\rho \hat{0}$; for all other balls just moved to β^+ , we change the label to $\rho \hat{1}$.
- (d) There is some child e -splitting node β , some $\rho \in \{0, 1\}^e$, and some $k \geq 1$, such that (β, ρ, k) is ready for definition at stage s .

For each such pair (β, ρ) , for the unique such k , letting $\alpha = \beta^-$, we define

$$f_{s+1}^{\alpha, \rho}(k) = \varphi_s(k) + 1.$$

Let $v = u_s^{\alpha, \rho}(k - 1)$ if $k > 1$, $v = 0$ otherwise. Let x_1, x_2, \dots , enumerate (in order) the elements of $S(\beta, \rho)_s$ that are greater than v . We let the A_s -use $u_{s+1}^{\alpha, \rho}(k)$ of the new computation be $x_{2k} + 1$.

We set $k_{s+1}^\rho(\beta) = k$.

- (e) If no case above applies, then s is declared to be a new balls stage.
 - (i) We let $H_{s+1} = H_s \cup A_{s+1}$, and remove any $x \in A_{s+1}$ from the machine.
 - (ii) We place any $x \in Q_{s+1} \setminus H_{s+1}$ not already on the machine at the root of the machine, and give it the empty string as a label.
 - (iii) For any e -splitting child node β , a child of some α , if $k_s^\rho(\beta) \downarrow = k$ but $f_{s+1}^{\alpha, \rho}(k) \uparrow$, then $k_{s+1}^\rho(\beta) \uparrow$.

Verification. We start with some observations on ball movements and labels, that follow from the instructions.

Lemma 3.8. *Let α be a node, s be a stage, and suppose that $x \in S(\alpha)_s$.*

- (a) *If $t > s$ and $x \in S(\beta)_t$, then $\beta \leq \alpha$ (in the Kleene-Brouwer ordering).*
- (b) *If $t > s$ and $x \in S(\alpha)_t$ (that is, x has not moved between stages s and t), then x 's label at stage s is the same as x 's label at stage t .*
- (c) *If α is an e -decision or e -splitting node (parent or child), then x 's label at stage s has length e .*

(d) If $t > s$ and $x \in Y(\alpha)_t$ then x 's label at stage t extends x 's label at stage s .

The proof of Lemma 2.8 gives:

Lemma 3.9. For all s , $Y(\lambda)_s = Q_s \setminus H_s$.

Lemma 3.10. If a ball x resides at a node α at some stage, then $|\alpha| \leq x + 2$.

Proof. If α is the root then this is immediate. Otherwise, let $\eta \preceq \alpha$ be the longest node which is the child of a decision node. Each node only receives balls that have already passed through their parent. Hence, there is a stage at which x was pulled by η . By Definition 3.3(ii), $x > |\eta|$; and $|\alpha| \leq |\eta| + 2$. \square

By Lemma 3.8(a) and Lemma 3.10, as in the proof of Lemma 2.9, we get:

Lemma 3.11. If $x \in \bigcup_s Y(\lambda)_s$ and $x \notin H$ then x is a permanent resident of some node.

Lemma 3.12. There are infinitely many new ball stages.

Proof. The point is that if not, then case (d) of the construction (defining new blocks) can happen at most finitely often. In more detail: suppose, for a contradiction, that s_0 is the last new balls stage. There is some stage $s_1 > s_0$ after which no balls are moved on the machine, or eliminated from the machine. If a new computation $f_{s+1}^{\alpha,\rho}(k)$ is defined at some stage $s > s_1$, then this computation is never made undefined. For every child β of α , for all $t > s$, if $S(\beta, \rho)_t \neq \emptyset$ then $k_t^\rho(\beta) \downarrow$ (at stage s , the fact that a new computation is defined, implies that there are no children of α which have balls, but insufficiently many to form a block: such balls would be pulled away earlier than stage s). Hence, no further values of $f^{\alpha,\rho}$ will be defined after stage s . \square

Since every element of A^c is added to the root of the tree at some stage, it follows that every $x \in H^c$ is a permanent resident of some node.

Defining the true path. We define the true path inductively. The root λ is declared to lie on the true path. Let α be a node on the true path.

- If α is a decision node, then the leftmost child β of α with $\bar{\ell}(\beta)$ unbounded is on the true path.
- If α is a parent splitting node, then the leftmost child β of α that releases balls infinitely often, lies on the true path.
- If α is a child splitting node, then its unique child is also on the true path.

As above, if α is a decision node, then as one of the statements $\psi(\alpha, D)$ is true, one of α 's children will lie on the true path. On the other hand, we will need to work to show that if α is a parent splitting node on the true path, then one of its children lies on the true path. For now, we do not know that the true path is infinite.

In fact, to show that a parent splitting node α which lies on the true path has a child on the true path, we will need to already know that the analogues of Lemmas 2.12 and 2.14 hold for α . Thus, we will need to prove both by simultaneous induction.

Recall (Definition 3.2) that we let

$$C_s = \{x \in Q_s : A_s \upharpoonright x + 1 = A \upharpoonright x + 1\},$$

the collection of balls $x \in Q_s$ which are A -true at stage s .

Proposition 3.13. *Suppose that a node α lies on the true path.*

- (a) *There is a stage after which for every node γ that lies to the left of α , γ does not eliminate any balls from A^c , nor is any ball from A^c moved to γ .¹⁷*
- (b) *Suppose that α is an e -splitting or decision node. Then for all $\rho \in \{0, 1\}^e$, for every k , there is some stage s such that $|Y(\alpha, \rho)_s \cap C_s| \geq k$.*

The proof of Proposition 3.13 will take some work. As discussed, we prove it by induction on the length of α . For now, we note:

Lemma 3.14. *Let α be a node; suppose that there is a stage after which no ball is moved to a node $\gamma <_L \alpha$. Then $Y(<_L \alpha)$ is finite.*

Proof. Every ball in $Y(<_L \alpha)$ is in A^c , and is at some stage moved to a node γ that lies to the left of α . By assumption, there are only finitely many such stages, and at each stage, only finitely many balls are on the machine. \square

Case I: the root. We start the inductive verification of Proposition 3.13 by verifying that it holds for the root. (a) is vacuous in this case; (b) is as in Lemma 2.14. We note that this is the only part of the proof in which we use the fact that we are using the true-stage enumeration of A .

Case II: decision nodes. Suppose that α is an e -decision node that lies on the true path, and suppose that Proposition 3.13 holds for α . Let β be α 's child that lies on the true path; we show that Proposition 3.13 holds for β as well. Let $D \subseteq \{0, 1\}^e$ such that $\beta = \alpha \hat{D}_n$ for some n . That is, $\rho \in D$ if and only if for all k there is some s such that $|Y(\alpha, \rho)_s \cap W_{e,s} \cap C_s| \geq k$.

To show that (a) holds for β , we only need a minor modification of the proof Lemma 2.12, reflecting the new second part of Definition 3.3(iii).

Let l be the maximum of $\lim_s \ell(\gamma)_s$, where γ is a child of α that lies to the left of β . Again, for any such γ and any $\varepsilon \geq \gamma$ that is a child of some decision node, $\lim_s \ell(\varepsilon)_s \leq l$. By Definition 3.3(vii), no such ε ever eliminates any $x \geq l$.

Let γ be a child of α that lies to the left of β , and suppose (by induction) that every child δ of α that lies to the left of γ eventually stops pulling any $x \in A^c$ (and so, from some stage, no ball from A^c is moved to any node that lies to the left of γ).

Let $\rho \in \{0, 1\}^e$. For a contradiction, suppose that (γ, ρ) pulls infinitely many balls from A^c . By our assumption, almost all such balls remain in $Y(\gamma)$. By Lemma 3.8(d), the balls pulled by (γ, ρ) that remain in $Y(\gamma)$ also remain in $Y(\gamma, \rho)$. Let s be a late stage at which $Y(\gamma, \rho)_s$ contains at least l many elements of A^c ; these balls will not be eliminated, so for all $t \geq s$, $|Y(\gamma, \rho)_t| \geq l$. Let $a = \max Y(\gamma, \rho)_s$. Then by induction on $t \geq s$ we show that $a = \max Y(\gamma, \rho)_t$: if this holds for t , then by (iii) of Definition 3.3, the only numbers pulled by (γ, ρ) are $< a$.

The proof that (b) holds for β is similar to that of Lemma 2.14, but here we use the new (iii) of Definition 3.3. Fix $\rho \in \{0, 1\}^e$. By (a), let s_0 be a stage after which no ball from A^c is moved to, or eliminated by, any node $\gamma <_L \beta$.

By Lemma 3.14, $Y(<_L \beta, \rho)$ is finite; let $m = |Y(<_L \beta, \rho)|$. If $s > s_0$ and $x \in Y(<_L \beta)_s \cap C_s$, then as x will not be later eliminated (and $x \notin A$), $x \in Y(<_L \beta)$. By Lemma 3.8(b), for all $s > s_0$, $Y(<_L \beta, \rho)_s \cap C_s \subseteq Y(<_L \beta, \rho)$. Hence, for all $s > s_0$, $|Y(<_L \beta, \rho)_s \cap C_s| \leq m$.

¹⁷Balls can move to γ without being pulled by γ , if γ is the child of a child splitting node, and balls are released by γ^- .

Let $k \in \mathbb{N}$. Suppose that $\rho \notin D$. By induction, there is some $s > s_0$ such that $|Y(\alpha, \rho)_s \cap C_s| \geq k + m + 3$ and such that $\ell(\beta)_s > k$. Let

$$R = \{x \in C_s \cap Y(\alpha, \rho)_s : x \notin Y(\leq \beta, \rho)_s \ \& \ x > |\alpha|\}.$$

Since $x \geq |\alpha| - 2$ for all $x \in Y(\alpha)_s$ (Lemma 3.10), $|R| \geq k - |Y(\beta, \rho)_s \cap C_s|$. So as in the maximal set construction, we will be done once we show that either $|Y(\beta, \rho)_s \cap C_s| \geq k$, or that every $x \in R$ is pullable by (β, ρ) at stage s (note that in the latter case, s will not be a new balls stage, so $C_s = C_{s+1}$). Suppose that $|Y(\beta, \rho)_s \cap C_s| < k$. The new part is that it is possible, in this case, that $|Y(\beta, \rho)_s| \geq \ell(\beta)_s$. If $|Y(\beta, \rho)_s| < \ell(\beta)_s$ then certainly all balls in R are pullable. If not, though, since $\ell(\beta)_s > k$, there must be some $z \in Y(\beta, \rho)_s$ which is not in C_s . However, $C_s \setminus H_s$ is an initial segment of the balls on the machine at stage s . So all balls in R are smaller than z . By the new part of Definition 3.3(iii), all balls in R are pullable by (β, ρ) at stage s .

The case $\rho \in D$ is the same, using the fact that as $\psi(\alpha, D)$ holds; there is some $s > s_0$ such that $\ell(\beta)_s > k$, and $|Y(\alpha, \rho)_s \cap C_s \cap W_{e,s}| \geq k + m + 3$. This completes the proof that Proposition 3.13 holds for β .

Case III: splitting nodes. For the rest of the proof of Proposition 3.13, let α be a parent e -splitting node that lies on the true path, and suppose that the proposition holds for α . We show that α has a child β on the true path, and that the proposition holds for the unique child β^+ of β (and hence also for β).

Until the end of the proof of Proposition 3.13, we fix a stage s_0 witnessing that Proposition 3.13(a) holds for α .

Lemma 3.15. *Let $\rho \in \{0, 1\}^e$; let s be a stage.*

- (a) *Let $k > 1$. If $f_s^{\alpha, \rho}(k) \downarrow$ then $f_s^{\alpha, \rho}(k-1) \downarrow$.*
- (b) *If β is a child of α and $k_s^\rho(\beta) \downarrow$, then $f_s^{\alpha, \rho}(k_s^\rho(\beta)) \downarrow$.*
- (c) *If β is a child of α and $k_s^\rho(\beta) \downarrow$, then for every child $\gamma < \beta$ of α , $k_s^\rho(\gamma) \downarrow$ and $k_s^\rho(\gamma) < k_s^\rho(\beta)$.*

Proof. (a) follows from the fact that when we define a new computation $f_{s+1}^{\alpha, \rho}(k)$, we have $f_s^{\alpha, \rho}(k-1) \downarrow$, and we set $u_{s+1}^{\alpha, \rho}(k) > u_s^{\alpha, \rho}(k-1)$. Also, we note that such a stage s is not a new balls stage, so $A_s = A_{s+1}$, so $u_s^{\alpha, \rho}(k-1) = u_{s+1}^{\alpha, \rho}(k-1)$.

(b) is by our stipulation that when an A -change causes a computation $f_s^{\alpha, \rho}(k)$ to become undefined at stage $s+1$, and $k = k_s^\rho(\beta)$, then we set $k_{s+1}^\rho(\beta) \uparrow$.

(c) follows from the previous two parts, and the fact that a new computation is always set up at the leftmost “free” child (a child with $k_s^\rho(\beta) \uparrow$); see the proof of Lemma 3.7. \square

Recall that we say that a computation $f_s^{\alpha, \rho}(k) \downarrow$ is A -correct if $A_s \upharpoonright u = A \upharpoonright u$, where $u = u_s^{\alpha, \rho}(k)$ is the use of the computation $f_s^{\alpha, \rho}(k)$. Also recall that $f_s^{\alpha, \rho}(k)$ is A -correct if and only if for all $t \geq s$, $f_t^{\alpha, \rho}(k) \downarrow$.

Lemma 3.16. *Let β be a child of α ; let $\rho \in \{0, 1\}^e$; let s be a stage. Suppose that $k = k_s^\rho(\beta) \downarrow$, and suppose that the computation $f_s^{\alpha, \rho}(k)$ was defined after stage s_0 , and is A -correct. Then $S(\beta, \rho)_s$ contains $2k$ many balls $x < u_s^{\alpha, \rho}(k)$.*

Proof. Let r be the stage at which the computation was defined. Let X be the set of $x \in S(\beta, \rho)_r$ such that $x < u_{r+1}^{\alpha, \rho}(k)$. Then by the construction, $|X| \geq 2k$, and by assumption that the computation is A -correct, $X \subseteq A^c$. By induction on $t \in [r, s]$ we see that $X \subseteq S(\beta, \rho)_t$. Suppose that this holds for t , and $t < s$. Since $t > s_0$,

no ball from X is pulled by a node to the left of α . If $\gamma < \beta$ is a child of α , and any ball from X is moved to $Y(\gamma)_{t+1}$, then $k_{t+1}^\rho(\gamma) \uparrow$; by Lemma 3.15, $k_{t+1}^\rho(\beta) \uparrow$. But then, since $f_t^{\alpha,\rho}(k)$ is A -correct, there is no stage $w > t$ at which $k_w^\rho(\beta) = k$, contradicting the hypothesis of this lemma. Hence, $X \subseteq S(\beta)_{t+1}$. \square

The following follows from the construction.

Lemma 3.17. *Let β be a child of α , and let $\rho \in \{0, 1\}^e$. Let $s \geq s_0$ be a stage.*

- (a) *Suppose that for all $t \geq s$, $k_t^\rho(\beta) \downarrow$. Then $f_s^{\alpha,\rho}(k_s^\rho(\beta))$ is A -correct, and for all $t \geq s$, $k_t^\rho(\beta) = k_s^\rho(\beta)$, and β does not release balls at stage t .*
- (b) *Suppose that $s \geq s_0$, $k_s^\rho(\beta) \downarrow$, and $f_s^{\alpha,\rho}(k_s^\rho(\beta))$ is A -correct. Suppose, further, that for all children $\gamma \leq \beta$ of α , for all $t \geq s$, γ does not release balls at stage t . Then for all $t \geq s$, $k_t^\rho(\beta) \downarrow$.*

Definition 3.18. Let β be a child of α , and let $\rho \in \{0, 1\}^e$. We write $k^\rho(\beta) \downarrow$ if $k_s^\rho(\beta) \downarrow$ for all but finitely many s .

The following is the main lemma.

Lemma 3.19. *Let β be a child of α , and suppose that for every every child $\gamma < \beta$ of α , for every $\rho \in \{0, 1\}^e$, $k^\rho(\gamma) \downarrow$. Then for all $\rho \in \{0, 1\}^e$, there are infinitely many stages t such that $k_t^\rho(\beta) \downarrow$ and $f_t^{\alpha,\rho}(k_t^\rho(\beta))$ is A -correct.*

An immediate consequence is the following:

Lemma 3.20. *Let β be a child of α , and suppose that for every every child $\gamma < \beta$ of α , for every $\rho \in \{0, 1\}^e$, $k^\rho(\gamma) \downarrow$. Then either:*

- (1) *For all $\rho \in \{0, 1\}^e$, $k^\rho(\beta) \downarrow$; or*
- (2) *β releases balls infinitely often, in fact, for all $\rho \in \{0, 1\}^e$ there are infinitely many t such that β releases balls at stage t , and $f_t^{\alpha,\rho}(k_t^\rho(\beta))$ is A -correct.*

Proof. Let $s_1 \geq s_0$ be a stage witnessing the assumption on the children $\gamma < \beta$ of α : for all $s \geq s_1$, for all $\rho \in \{0, 1\}^e$, $k_s^\rho(\gamma) \downarrow$.

Let $\rho \in \{0, 1\}^e$. Let $s \geq s_1$ be a stage such that $k_s^\rho(\beta) \downarrow$, and $f_s^{\alpha,\rho}(k_s^\rho(\beta))$ is A -correct. If β does not release any balls after stage s , then by Lemma 3.17(b), stage s shows that $k^\rho(\beta) \downarrow$. So if β releases balls only finitely many times, (1) holds for β .

Suppose that β releases balls infinitely often; let t be the least stage $\geq s$ at which β releases balls. The assumption that $f_s^{\alpha,\rho}(k_s^\rho(\beta))$ is A -correct, and $s \geq s_1$, imply that $k = k_t^\rho(\beta) = k_s^\rho(\beta)$ and that $f_t^{\alpha,\rho}(k) \downarrow$ is A -correct, so t is one of the stages as required for (2). \square

Proof of Lemma 3.19. Let $s_1 \geq s_0$ be a stage witnessing the assumption on the children $\gamma < \beta$ of α . Fix some $\rho \in \{0, 1\}^e$. Let $s_2 \geq s_1$ be any stage. We show that there is some $t \geq s_2$ such that $k_t^\rho(\beta) \downarrow$ and $f_t^{\alpha,\rho}(k_t^\rho(\beta))$ is A -correct. Suppose, for a contradiction, that this is not the case. In other words:

\otimes_1 : If $s \geq s_2$ and $k_s^\rho(\beta) \downarrow = k$, then there is some $t > s$ such that $f_t^{\alpha,\rho}(k) \uparrow$.

Let m be the least such that there is some $s \geq s_2$ such that $f_s^{\alpha,\rho}(m) \uparrow$. For all $k < m$, $f_{s_2}^{\alpha,\rho}(k) \downarrow$ is A -correct. Hence, our assumption for contradiction implies:

\otimes_2 : For all $s \geq s_2$, either $k_s^\rho(\beta) \uparrow$, or $k_s^\rho(\beta) \geq m$.

We claim:

\otimes_3 : There are infinitely many stage s at which $f_s^{\alpha,\rho}(m) \uparrow$.

Suppose, for a contradiction, that this is not the case. Thus, $f^{\alpha,\rho}(m) \downarrow$. Let r be least such that $f_r^{\alpha,\rho}(m)$ is A -correct, i.e., the correct computation $f^{\alpha,\rho}(m)$ is defined at stage $r - 1$. By the choice of m , $r > s_2$. The computation is established at some child γ of α : $k_r^\rho(\gamma) = m$. Since $r > s_1$, $\gamma \geq \beta$. By Lemma 3.15, if $\gamma > \beta$ then $k_r^\rho(\beta) \downarrow < m$, contradicting \otimes_2 . Hence, $\gamma = \beta$. But then, by \otimes_1 , there is some $s > r$ such that $f_s^{\alpha,\rho}(m) \uparrow$, giving the desired contradiction that establishes \otimes_3 .

We also observe:

\otimes_4 : If $s \geq s_2$ and $f_s^{\alpha,\rho}(m) \uparrow$ then $k_s^\rho(\beta) \uparrow$.

This follows from Lemma 3.15 and \otimes_2 : if $k = k_s^\rho(\beta) \downarrow$ then $f_s^{\alpha,\rho}(k) \downarrow$ and $k \geq m$, so $f_s^{\alpha,\rho}(m) \downarrow$.

There are only finitely many stages s such that $k_s^\rho(\beta) = m$ and β releases balls at stage s (this is the main point of using the certification process). To see this, let s be such that $k_s^\rho(\beta) = m$ and β releases balls at stage s . Let r be the stage at which the computation $f_s^{\alpha,\rho}(m)$ was defined. Then $\varphi_r(m) < f_s^{\alpha,\rho}(m) < \varphi_s(m)$, in particular, $\varphi_r(m) \neq \varphi_s(m)$. Thus, there is at most one such stage s after the last stage at which $\varphi_t(m)$ changes.

Let $s_3 \geq s_2$ be a stage after which there are no such stages s ; by \otimes_3 , assume that $f_{s_3}^{\alpha,\rho}(m) \uparrow$.

\otimes_5 : For all $s \geq s_3$, either $f_s^{\alpha,\rho}(m) \uparrow$ or $k_s^\rho(\beta) = m$.

We prove \otimes_5 by induction on $s \geq s_3$. By assumption, this holds for $s = s_3$. Let $s \geq s_3$. First, suppose that $f_s^{\alpha,\rho}(m) \uparrow$. By \otimes_4 , $k_s^\rho(\beta) \uparrow$. Hence, if a new computation $f_{s+1}^{\alpha,\rho}(m)$ is defined at stage s , it is established at β , setting $k_{s+1}^\rho(\beta) = m$. If not, then $f_{s+1}^{\alpha,\rho}(m) \uparrow$. Next, suppose that $k_s^\rho(\beta) = m$. Since $s \geq s_3$, β does not release any balls at stage s . Hence, either $k_{s+1}^\rho(\beta) = m$, or $f_{s+1}^{\alpha,\rho}(m) \uparrow$.

Putting \otimes_5 and the choice of s_3 together, we obtain:

\otimes_6 : β does not release balls at any stage $s \geq s_3$.

By choice of s_0 and s_1 , after stage s_3 , no ball from A^c is moved to β^+ , or from β^+ to any node to its left. So for all $s \geq s_3$, $Y(\beta^+, \rho)_s = Y(\beta^+, \rho)$, so $Y(\beta^+, \rho)$ is finite. Let $N = |Y(\beta^+, \rho)|$. Let $v = u^{\alpha,\rho}(m - 1)$ if $m > 1$, $v = 0$ otherwise.

By the assumption that Proposition 3.13(b) holds for α , let $s \geq s_3$ be a stage at which $|Y(\alpha, \rho)_s \cap C_s| \geq N + 2m + v$. Let X be the set of $x \in Y(\alpha, \rho)_s \cap C_s$ such that $x \notin Y(\beta^+)_s$ and $x > v$.

Then $|X| \geq 2m$, and for all $t \geq s$, $X \subseteq Y(\alpha, \rho)_t \setminus Y(\beta^+)_t$. By \otimes_3 and \otimes_4 , let $t > s$ be a stage such that $k_t^\rho(\beta) \uparrow$. Then by Definition 3.5, either $X \subseteq S(\beta, \rho)_t$, or balls are pulled by β at stage t , resulting in $X \subseteq S(\beta, \rho)_{t+1}$. In either case, we obtain a stage $r \geq s_3$ at which $f_r^{\alpha,\rho}(m) \uparrow$, $k_r^\rho(\beta) \uparrow$, and $X \subseteq S(\beta, \rho)_r$. So at stage r we define a new computation $f_{r+1}^{\alpha,\rho}(m)$ and set $k_{r+1}^\rho(\beta) = m$. The use $u_{r+1}^{\alpha,\rho}(m)$ is defined to be $x + 1$ for some $x \in X$. Since $X \subseteq C_r$, this shows that $f_{r+1}^{\alpha,\rho}(m)$ is A -correct. This contradicts \otimes_3 , which finishes the proof of Lemma 3.19. \square

Lemma 3.21. *Some child of α lies on the true path.*

Recall that this means that some child of α releases balls infinitely often.

Proof. Suppose, for a contradiction, that no child of α releases balls infinitely often. By induction on the children of α , from left to right, we see, using Lemma 3.20, that for every child β of α , for every $\rho \in \{0, 1\}^e$, $k^\rho(\beta) \downarrow$. Fixing ρ , since the map $\beta \mapsto k^\rho(\beta)$ is injective (Lemma 3.15), we see that $f^{\alpha,\rho}$ is total.

Since each $f^{\alpha,\rho}$ is A -computable, φ dominates $\max_\rho f^{\alpha,\rho}$. Say this domination starts at k^* . Let β be a child of α that lies sufficiently to the right, so that for all $\rho \in \{0,1\}^e$, $k^\rho(\beta) \geq k^*$. Let s be sufficiently late so that for each ρ , for all $t \geq s$, $k_t^\rho(\beta) \downarrow = k^\rho(\beta)$, and $\varphi_s(k_t^\rho) > f^{\alpha,\rho}(k^\rho(\beta))$. But then, β is certified at each stage $t \geq s$, and so, will release balls – contradicting Lemma 3.17. \square

Let β be the child of α that lies on the true path.

Lemma 3.22. *Proposition 3.13 holds for both β and β^+ .*

Proof. By Lemma 3.20, for every child $\gamma < \beta$ of α , for all ρ , $k^\rho(\gamma) \downarrow$. By Lemma 3.17, eventually, each such child stops pulling any balls (whether from A^c or not), and does not release any balls. Thus, for each such γ , only finitely many balls ever enter $Y(\gamma)$, and each such ball is eventually removed from the machine, or stops moving. This gives part (a) of Proposition 3.13 for β^+ (and hence for β as well).

Part (b) follows from Lemma 3.20 as well. Fix $\rho \in \{0,1\}^e$. Let t be a stage. If $k_t^\rho(\beta) \downarrow = k$, $f_t^{\alpha,\rho}(k)$ is A -correct, and β releases balls at stage t , then by Lemma 3.16, $|S(\beta^+, \rho)_{t+1} \cap C_{t+1}| \geq 2k$. By construction, in fact, for each $i = 0, 1$, $|S(\beta^+, \rho^i)_{t+1} \cap C_{t+1}| \geq k$.

If $t' > t$, $k_{t'}^\rho(\beta) \downarrow = k'$, and $f_{t'}^{\alpha,\rho}(k')$ is A -correct, then $k' > k$, showing that these numbers k go to ∞ . \square

Corollary 3.23. *The true path is infinite, and Proposition 3.13 holds for every α on the true path.*

The rest. The rest is now straightforward. For each $\rho \in 2^{<\omega}$, let

$$Z(\rho)$$

be the collection of balls $x \notin H$ whose permanent label extends ρ .

Lemma 3.24. *Let α on the true path be an e -decision or splitting node. For every $\rho \in \{0,1\}^e$, $Y(\alpha, \rho) \neq \emptyset$.*

Proof. Very much like the proof of Lemma 2.16, except that replacing true stages by C_s actually makes things slightly simpler. We may assume that α is an e -decision node. Let s_0 be a stage after which nodes that lie to the left of α do not eliminate any balls, or pull any balls in A^c . Let

$$x = \min \bigcup_{s > s_0} (C_s \cap Y(\alpha, \rho)_s).$$

x exists by Proposition 3.13. Say $s_1 > s_0$ and $x \in C_{s_1} \cap Y(\alpha, \rho)_{s_1}$. By induction on $t \geq s_1$ we see that $x = \min Y(\alpha, \rho)_t$. For $t = s_1$ this is because C_{s_1} is an initial segment of Q_{s_1} . Suppose that $x = \min Y(\alpha, \rho)_t$. Then x is not eliminable by any node $\gamma \geq \alpha$, and so $x \in Y(\alpha, \rho)_{t+1}$. Let $y < x$. If $y \in Y(\alpha, \rho)_{t+1}$, then as $C_{s_1} \subseteq C_{t+1}$, we have $y \in C_{t+1}$, contradicting the minimality of x . \square

Corollary 3.25. *Let α on the true path be an e -decision or splitting node. For every $\rho \in \{0,1\}^e$, $Y(\alpha, \rho)$ is infinite.*

It follows that each $Z(\rho)$ is infinite.

Lemma 3.26. *For every α on the true path, $Y(\alpha) =^* H^c$.*

Proof. Like the proof of Lemma 2.18. The new part is when α is a child e -splitting node; we need to show that every ball that lies to the right of α but below α^- is eventually pullable by α . This follows from the fact that α releases balls infinitely often; if α releases balls at stage s , then $k_{s+1}^\rho(\alpha) \uparrow$ for each ρ , and then every such ball x is pullable by α at stage $s + 1$. \square

As a result, for all ρ , if α is an e -decision or splitting node on the true path, then $Z(\rho) =^* Y(\alpha, \rho)$.

Lemma 3.27. *For every ρ , $H \cup Z(\rho)$ is c.e.*

Proof. Let α be an e -decision node on the true path, where $|\rho| = e$. Then $Y(\alpha, \rho)$ is c.e.: from some stage onwards, any ball outside H which enters $Y(\alpha)$ with label ρ , will stay in $Y(\alpha, \rho)$. \square

Lemma 3.28. *For every ρ , $Z(\rho) =^* Z(\rho^{\hat{0}}) \cup Z(\rho^{\hat{1}})$, and $Z(\rho^{\hat{0}}) \cap Z(\rho^{\hat{1}}) = \emptyset$.*

Proof. If α is an e -decision node on the true path with $e = |\rho| + 1$, then $Y(\alpha, \rho) = Y(\alpha, \rho^{\hat{0}}) \cup Y(\alpha, \rho^{\hat{1}})$. \square

Lemma 3.29. *For every e , $W_e \cap H^c$ is the union of finitely many sets $Z(\rho)$, up to finite difference.*

Proof. Like the proof of Lemma 2.20. Let α on the true path be a child of an e -decision node. Then for all $\rho \in \{0, 1\}^e$, either $Y(\alpha, \rho) \subseteq W_e$, or $Y(\alpha, \rho) \cap W_e =^* \emptyset$. The result follows since $Z(\rho) =^* Y(\alpha, \rho)$ for each such ρ , and $Z(\rho)$ for $\rho \in \{0, 1\}^e$ partition H^c (up to finite difference). \square

Corollary 3.30. *H is atomless hyperhypersimple.*

4. OTHER BOOLEAN ALGEBRAS

In this section, we explain how to modify the proof of Theorem 1.2 to show:

Theorem 4.1. *Let A be low₂ and coinfinite. For any Σ_3 -Boolean algebra B there is some c.e. $H \supseteq A$ such that $\mathcal{L}^*(H) \cong B$.*

Let B be a Σ_3 -Boolean algebra. This means that it is the quotient of the (standard computable copy of the) atomless Boolean algebra by a Σ_3 ideal. An equivalent characterisation is in terms of Boolean algebras generated by trees. Recall that for a tree $T \subseteq 2^{<\omega}$ we defined the Boolean algebra $B(T)$, generated from the elements of T according to the rules: (i) every $\tau \in T$ is the join of its children; and (ii) if $\sigma, \tau \in T$ are incomparable then $\sigma \wedge \tau = 0_{B(T)}$. We noted that this implies that if τ is non-extendible on T , then $\tau = 0_{B(T)}$ (and the atoms of $B(T)$ are τ where τ isolates a unique path of T). Note that there are other ways of producing Boolean algebras from trees, for example, ones in which leaves of the tree are atoms. However, the advantage of the representation that we chose is:

- The Σ_3 Boolean algebras are precisely the algebras $B(T)$, where T is Δ_3 . (If we used the other representation, we would need Σ_3 trees). The idea of the modified construction is to add nodes that decide, for each $\rho \in \{0, 1\}^e$, whether $\rho \in T$ or not, again utilising Δ_3 guessing. A child β of such α that guesses that $\rho \notin T$ will attempt to enumerate all of $Y(\alpha, \rho)$ into H . The technique is identical to eliminating balls to decide W_e on $Y(\alpha, \rho)$.

The construction. Most of the construction is as above, so we explain the new ingredients. We now have four kinds of nodes:

- e -tree nodes, of length $4e$;
- e -decision nodes, of length $4e + 1$;
- parent and child e -splitting nodes, of lengths $4e + 2$ and $4e + 3$.

For an e -tree node α and $E \subseteq \{0, 1\}^e$, the statement $\psi(\alpha, E)$ is:

$$E = T \cap \{0, 1\}^e.$$

This is a finite Boolean combination of Δ_3 statements. Again the children are $\alpha \hat{E}_n$ for $E \subseteq \{0, 1\}^e$ and $n < \omega$. We define $\bar{\ell}(\alpha \hat{E}_n) = \bar{\ell}(\psi(\alpha, E), n)$ as above.

Definition 4.2. Let $\beta = \alpha \hat{E}_n$ be a child of an e -tree node α , and let $\rho \in \{0, 1\}^e$. We say that a ball x is *pullable* by (β, ρ) at a stage s if:

- (i) $x \in Y(\alpha, \rho)_s \setminus Y(\leq \beta, \rho)_s$;
- (ii) $\rho \in E$; and
- (iii) either:
 - $|Y(\beta, \rho)_s| < \ell(\beta)_s$, or
 - $Y(\beta, \rho)_s \neq \emptyset$ and $x < \max Y(\beta, \rho)_s$.

We say that x is *eliminable* by (β, ρ) at stage s if:

- (iv) $x \in Y(\alpha, \rho)_s \setminus Y(\leq \beta, \rho)_s$;
- (v) $\rho \notin E$;
- (vi) $x \neq \min Y(\alpha, \rho)_s$; and
- (vii) $x < \ell(\alpha)_s$.

We need to modify Definition 3.5(c) so that we require certification only for those $\rho \in \{0, 1\}^e$ that are guessed to be on the tree T . If α is an e -decision, e -splitting, or $(e + 1)$ -tree node, then we let

$$E(\alpha)$$

denote the set $E \subseteq \{0, 1\}^e$ such that for some n , $\varepsilon \hat{E}_n \leq \alpha$, where ε is the e -tree node preceding α . We let $E(\lambda) = \{\lambda\}$. Then in the new version of Definition 3.5(c), we require that the conditions hold for those $\rho \in E(\beta)$.

The rest of the construction follows the atomless hyperhypersimple construction above, verbatim.

The verification. For the verification, we only need to replace the second part of Proposition 3.13. The new version of Proposition 3.13(b) replaces $\{0, 1\}^e$ by $E(\alpha)$:

- If α lies on the true path, then for all $\rho \in E(\alpha)$, for all k , there is some s such that $|Y(\alpha, \rho)_s \cap C_s| \geq k$.

The rest of the verification follows without change, showing that if $\rho \in T$ then $Z(\rho)$ is infinite, while if $\rho \notin T$ then $Z(\rho)$ is finite.

REFERENCES

- [Cho95] Peter Cholak. *Automorphisms of the lattice of recursively enumerable sets*, volume 113 of *Mem. Amer. Math. Soc.* American Mathematical Society, 1995.
- [DG20] Rod Downey and Noam Greenberg. *A hierarchy of Turing degrees*, volume 206 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 2020. A transfinite hierarchy of lowness notions in the computably enumerable degrees, unifying classes, and natural definability.
- [DJS13] R. Downey, C. Jockusch, and E. Schupp. Asymptotic density and computably enumerable sets. *Journal of Mathematical Logic*, 13:02, 2013.

- [Fri58] R. Friedberg. Three theorems on recursive enumeration. *The Journal of Symbolic Logic*, 23:309–316, 1958.
- [HS96] Leo Harrington and Robert Soare. The Δ_3^0 -automorphism method and noninvariant classes of degrees. *J. Amer. Math. Soc.*, 9(3):617–666, 1996.
- [Lac68a] Alistair Lachlan. Degrees of recursively enumerable sets with no maximal supersets. *Journal of Symbolic Logic*, 33:431–443, 1968.
- [Lac68b] Alistair Lachlan. On the lattice of recursively enumerable sets. *Trans. Amer. Math. Soc.*, 130:1–37, 1968.
- [Lac76] Alastair Lachlan. A recursively enumerable degree which will not split over all lesser ones. *Annals of Mathematical Logic*, 9(4):307–365, 1976.
- [Maa83] W. Maass. Characterization of the recursively enumerable sets with supersets effectively isomorphic to all recursively enumerable sets. *Transactions of the American Mathematical Society*, 279:311–336, 1983.
- [Mar66] D. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 12:295–310, 1966.
- [Pos44] Emil Post. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society*, 50:284–316, 1944.
- [Rob71] R. W. Robinson. Jump restricted interpolation in the recursively enumerable degrees. *Annals of Mathematics. Second Series*, 93:586–596, 1971.
- [Sho76] Joseph Shoenfield. Degrees of classes or r.e. sets. *Journal of Symbolic Logic*, 41:695–696, 1976.
- [Soa82] Robert I. Soare. Automorphisms of the lattice of recursively enumerable sets. II. low sets. *Annals of Mathematical Logic*, 22:69–107, 1982.
- [Soa87] Robert I. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.
- [SS90] R. Shore and T. Slaman. Working below a low₂ recursively enumerable degree. *Archive for Mathematical Logic*, 29:201–211, 1990.
- [SS93] R. Shore and T. Slaman. Working below a high recursively enumerable degree. *Journal of Symbolic Logic*, 58:824–859, 1993.
- [Tur39] Alan Turing. Systems of logic based on ordinals. *Proceedings of the London Math. Society*, 45:154–222, 1939.

Peter Cholak MATHEMATICS DEPARTMENT, UNIVERSITY OF NOTRE DAME, SOUTH BEND, IN 46556 USA

Email address: `Peter.Cholak.1@nd.edu`

Rodney Downey and Noam Greenberg SCHOOL OF MATHEMATICS AND STATISTICS, VICTORIA UNIVERSITY, P.O. BOX 600, WELLINGTON, NEW ZEALAND.

Email address: `rod.downey@vuw.ac.nz`, `noam.greenberg@vuw.ac.nz`