

On n -Tardy Sets

Peter A. Cholak^{a,1,*}, Peter M. Gerdes^{a,2}, Karen Lange^{a,3}

^a*Department of Mathematics
University of Notre Dame
Notre Dame, IN 46556-5683*

Abstract

Harrington and Soare introduced the notion of an n -tardy set. They showed that there is a nonempty \mathcal{E} property $Q(A)$ such that if $Q(A)$ then A is 2-tardy. Since they also showed no 2-tardy set is complete, Harrington and Soare showed that there exists an orbit of computably enumerable sets such that every set in that orbit is incomplete. Our study of n -tardy sets takes off from where Harrington and Soare left off. We answer all the open questions asked by Harrington and Soare about n -tardy sets. We show there is a 3-tardy set A that is not computed by any 2-tardy set B . We also show that there are nonempty \mathcal{E} properties $Q_n(A)$ such that if $Q_n(A)$ then A is properly n -tardy.

Keywords: c.e. sets, r.e. sets, automorphisms, n -tardy sets

2010 MSC: 03D25, 03D30

1. Introduction

Let \mathcal{E} denote the structure of c.e. sets under the language of inclusion. Understanding the interplay between computability and definability in \mathcal{E} is

*Corresponding author

Email addresses: `Peter.Cholak.1@nd.edu` (Peter A. Cholak),
`gerdes@invariant.org` (Peter M. Gerdes), `klange1@nd.edu` (Karen Lange)
URL: <http://www.nd.edu/~cholak> (Peter A. Cholak), <http://invariant.org>
(Peter M. Gerdes), <http://www.nd.edu/~klange1/> (Karen Lange)

¹Cholak was partially supported by NSF DMS-0652669 and NSF-DMS-0800198.

²Gerdes was partially supported by NSF EMSW21-RTG-0739007 and EMSW21-RTG-0838506.

³Lange was partially supported by NSF DMS-0802961.

a longstanding area of research in classical computability theory. In 1944 [8], Post set out to find an incomplete noncomputable c.e. set, i.e., a noncomputable c.e. set that does not have the degree of the halting problem K . He defined several properties of c.e. sets (such as *simplicity*) in the hope that no c.e. set satisfying one of these properties could be complete. All of the properties he suggested failed to satisfy this condition, but many of them are definable in \mathcal{E} . Although Friedberg and Muchnik [7, 2] famously obtained an incomplete noncomputable c.e. set using a priority argument, a natural question is whether there exists an \mathcal{E} -definable nontrivial property Q such that if $Q(A)$ holds, then A is an incomplete noncomputable c.e. set. Harrington and Soare produced such a property Q in [4], and they also described an \mathcal{E} -definable property that guarantees completeness (See [9], p. 339 and [3]). These results are part of work by many towards the following general goal.

Question 1.1. *Characterize what sets are and are not automorphic to a complete set.*

Harrington and Soare showed that all sets that satisfy Q are *2-tardy* [6], a slowness condition that we describe, along with the conditions *n-tardy* and *very tardy*, in §1.2. The very tardy sets, by definition, are those that are not almost prompt, and all complete sets are prompt. All *n-tardy* sets are very tardy and, hence, incomplete. Thus, any A for which $Q(A)$ holds is not automorphic to a complete set. On the other hand, Harrington and Soare [5], building on work of the first author, Downey, and Stob [1], proved that every almost prompt set (i.e., every not very tardy set) is automorphic to a complete set. Thus, in order to work towards answering Question 1.1, we explore the very tardy sets and their orbits from the perspectives of computability and definability. We begin by defining the almost prompt sets.

1.1. Almost prompt

Definition 1.2. A set X is *n-c.e.* iff there is a computable sequence of c.e. sets $\{X_i : 1 \leq i \leq n\}$ such that

$$X = (X_1 - X_2) \cup \dots \cup (X_{n-2} - X_{n-1}) \cup X_n \text{ if } n \text{ is odd, and}$$

$$X = (X_1 - X_2) \cup \dots \cup (X_{n-1} - X_n) \text{ if } n \text{ is even.}$$

The sequence of sets $\{X_i : 1 \leq i \leq n\}$ is an *n-c.e.* presentation of X . Such a sequence can be used to give a stagewise approximation of X : (for the case

when n is even)

$$X_s = (X_{1,s} - X_{2,s}) \cup \dots \cup (X_{n-1,s} - X_{n,s})$$

Such a sequence is denoted X_e^n , and $X_{e,s}^n$ denotes the stagewise approximation given by the sequence.

Definition 1.3 (Definition 11.3 of [5]). Let A be a c.e. set and $\{A_s\}$ be an enumeration of A . The set A is *almost prompt* iff there is a nondecreasing function $p(s)$ such that for all n and all e

$$(1.4) \quad X_e^n = \bar{A} \implies (\exists x)(\exists s)[x \in X_{e,s}^n \wedge x \in A_{p(s)}].$$

Harrington and Soare [5] showed that this definition is robust. That is, if Equation (1.4) holds for some enumeration of A , it holds for all enumerations of A (see [5, Theorem 11.4]). They also proved that any c.e. set of prompt degree is almost prompt (see [5, Theorem 11.7]); thus, the notion of almost prompt generalizes the notion of prompt. They also showed that almost prompt sets are ubiquitous in the following sense.

Theorem 1.5 (Harrington, Soare, Theorem 11.12 [5]). *There are almost prompt sets of every c.e. degree.*

Moreover, they showed that there are tardy (i.e., not of prompt degree) sets A such that every degree Turing above A is almost prompt, (see [5, Theorem 11.8]) and that the join of an almost prompt set and any computably enumerable set is almost prompt (see [5, Theorem 11.11]).

In order to show Theorem 1.5, Harrington and Soare proved that every low simple set is almost prompt (see [5, Theorem 11.10]). They left the following question open:

Question 1.6 (Question 1 of [5]). *If A is low₂ and simple, is A almost prompt?*

We provide a negative answer to Question 1.6 in §5, but we first focus on particular classes of sets that are not almost prompt.

1.2. Very tardy and n -tardy sets

A degree is *tardy* if it is not a prompt degree. A set is *very tardy* if it is not almost prompt. (Note that being very tardy is a property of sets and does not readily extend to degrees.) Since the definition of almost prompt is robust, we have the following equivalent definition.

Definition 1.7. Let A be c.e. and $\{A_s\}$ be an enumeration of A . The set A is *very tardy* iff A is not almost prompt iff for every nondecreasing computable function $p(s)$ there is an n and an e such that

$$(1.8) \quad X_e^n = \bar{A} \implies (\forall x)(\forall s)[x \in X_{e,s}^n \implies x \notin A_{p(s)}].$$

Moreover, A is *n -tardy* iff there is a single n that works for all such functions $p(s)$, and A is *properly n -tardy* if A is n -tardy but not $n - 1$ -tardy.

As described in the introduction, Harrington and Soare proved the following theorem.

Theorem 1.9 (Harrington and Soare [4]). *There exists an \mathcal{E} -definable non-trivial property Q such that if $Q(A)$ holds, then A is not automorphic to a complete set.*

More specifically, they showed the property Q describes a subset of the 2-tardy sets. We need a few definitions in order to define this subset.

Definition 1.10. 1. Let $A \subset_\infty C$ denote that $A \subset C$ and $C - A$ is infinite.
2. A subset A is a *major* subset of C if $A \subset_\infty C$ and for all e ,

$$\bar{C} \subseteq W_e \implies \bar{A} \subseteq^* W_e.$$

3. A subset $A \subset C$ is a *small* subset of C (written $A \subset_s C$) if $A \subset_\infty C$ and for all X and Y ,

$$X \cap (C - A) \subseteq Y \implies (\exists Z)_{Z \subseteq X}[Z \supseteq (X - C) \ \& \ (Z \cap C) \subseteq Y].$$

4. If A is both a small subset and a major subset of C , we call A a *small major* subset of C and write $A \subset_{sm} C$.

Theorem 1.11 (Harrington and Soare [6]). $Q(A) \iff (\exists C)[A \subset_{sm} C \ \& \ A \text{ is } 2\text{-tardy}]$.

Harrington and Soare used this characterization to show that any A satisfying $Q(A)$ is not automorphic to a complete set.

Definition 1.12. The *orbit* of A , denoted by $[A]$, is the set of c.e. sets B such that there exists an automorphism Ψ of \mathcal{E} sending A to B .

If A satisfies $Q(A)$ and there is an automorphism Ψ of \mathcal{E} , then $Q(\Psi(A))$ holds as well. In other words, Q holds of any element in $[A]$. Since Q holds of all sets in $[A]$, all sets in $[A]$ are 2-tardy and therefore incomplete. Thus, if A satisfies $Q(A)$, A is not automorphic to a complete set.

In §3, we define nontrivial properties \hat{Q}_n that generalize Q . In Theorem 3.3, we show that if $\hat{Q}_n(A)$ holds, then A is n -tardy and $\neg\hat{Q}_i(A)$ holds for all $i < n$. In Theorem 4.1, we show that there is some properly n -tardy set A_n for which $\hat{Q}_n(A_n)$ holds. Thus, the collection $\{[A_n]\}_{n \in \omega}$ witness that the c.e. sets that are not automorphic to a complete set break into countably many disjoint orbits.

1.3. Codeable sets

In [6], Harrington and Soare also explore the connection between tardiness and what sets X are coded in every nontrivial orbit in the following sense.

Definition 1.13 ([6] Definition 1.3). 1. We say X is *coded in the orbit of* A , denoted $X \leq_T [A]$, if $X \leq_T B$ for some $B \in [A]$.
2. We say X is *codeable* if for every noncomputable set A , $X \leq_T [A]$.

Harrington and Soare obtain the following characterization of the codeable sets by using the Δ_3^0 -automorphism method they developed in [3].

Theorem 1.14 (Harrington Soare [6] Corollary 1.8). *A set is codeable iff $X \leq_T D$ for some D satisfying $Q(D)$.*

Using Theorem 1.14, Harrington and Soare obtain the following simple corollary.

Corollary 1.15 (Harrington Soare [6] Corollary 1.9). *If S has prompt degree, then S is not codeable.*

Harrington and Soare in fact show that a set is codeable iff $X \leq_T D$ for some 2-tardy D ; Theorem 1.14 only uses the fact that if $Q(D)$ holds, then D is 2-tardy. Thus, the ability to code in the above sense is more connected to enumeration speed than degree-theoretic content. Given this observation, it is natural to wonder whether all very tardy sets are codeable. Harrington and Soare asked a more specific version of this problem:

Question 1.16 (Harrington Soare [5] Question 1). *Are all 3-tardy sets codeable?*

By Theorem 1.14, this is equivalent to the following question.

Question 1.17. *If A is 3-tardy, does there exist a 2-tardy set B such that $A \leq_T B$?*

Let A be 2-tardy. If $A_0 \sqcup A_1 = A$ is a nontrivial split, then each of the A_i are 3-tardy. Given $p(s)$ and X_e^2 such that Equation (1.8) holds for A , then $p(s)$ and $X_e^3 = (X_{e_1}^2 - X_{e_2}^2) \sqcup A_i$ witnesses Equation (1.8) holds for A_i , where $\bar{0} = 1$ and $\bar{1} = 0$. Prior to this work, it was unknown whether every 3-tardy is the split of a 2-tardy. If this was the case, then clearly every 3-tardy would be computable from the 2-tardy of which it is a split, and hence would be codeable. In §2, we show that not all 3-tardy sets are splits of 2-tardy sets. In fact, we answer Question 1.17 negatively. We show that there exists a 3-tardy set that is not computed by any 2-tardy. Hence, not all 3-tardy (and very tardy) sets are codeable.

2. A 3-tardy not computed by any 2-tardy

We devote this section to constructing a 3-tardy set A not computed by any 2-tardy set. Hence, A is non-codeable.

Theorem 2.1. *There is a 3-tardy set A such that for all 2-tardy sets B , $A \not\leq_T B$.*

Proof. We will construct A . Our construction style will be a pinball machine laid out on top of a tree. Here our tree will be $3^{<\omega}$. Since balls move downward (gravity) in this case we want to think of our tree as growing upward. As always, we are most concerned about the action of the pinball machine along the true path. We will have an approximation f_s of the true path f , such that $f = \liminf_s f_s$.

The approximation to the true path, f_s , will help determine the movement of the balls (integers) on the pinball machine. Balls will be placed on the pinball machine by a node $\alpha \subset f_s$ at stage s only when we wish to put them into A . At stage s , all balls x on the machine will be located at some node $\alpha(x, s)$. If $\alpha(x, s) = \lambda$ (λ is the empty node), we put x into A_{s+1} and remove x from the machine at stage $s + 1$. So when a ball x is on the machine our apparent goal is to move x downward and into A . At some point later, we will *sometimes* change our mind and remove balls from the machine, preventing them from going into A . If $f_{s+1} <_L \alpha(x, s)$ we will also remove x

from the machine at stage $s + 1$ and never use it again. At stage $s + 1$, we are free to place any ball $x \leq s + 1$ that has never been used on the machine. However, we must ensure that for all s , if $\alpha(x, s) \downarrow$ then, for all t such that $x \leq t \leq s$, $\alpha(x, s) \leq_L f_t$. The action to ensure this goes on at every stage in the background.

Our next goal is to make A a 3-tardy set. This means that balls must enter A very slowly. We have to meet the following requirements:

$$\mathcal{N}_e: \varphi_e \text{ total} \implies (\exists X_e^3)(X_e^3 = \bar{A} \wedge (\forall x)(\forall s)[x \in X_{e,s}^3 \implies x \notin A_{\varphi_e(s)}])$$

In general, the way to meet \mathcal{N}_e is to ensure that for all balls x there is a stage s_1 at which we put x into A_{s_1} or X_{e_1, s_1}^3 . Now if a ball x in X_{e_1, s_1}^3 wants to enter A at stage $s_2 > s_1$ we must put x into X_{e_2, s_2}^3 . Then we wait until a stage s_3 such that $\varphi_{e, s_3}(s_2) \downarrow$. If such a stage s_3 exists then we *must* eventually put x into A or $X_{e_3}^3$. If a ball x is in X_{e_2, s_2}^3 and we remove it from the machine at stage s_4 , we will put x into $X_{e_3}^3$ at stage s_4 . If $\varphi_{e, s_3}(s_2) \uparrow$ then φ_e is not total and the requirement is satisfied.

In the tree construction, we will use node γ to meet \mathcal{N}_e . We will label the 3-c.e. set constructed at γ , as X_γ^3 rather than X_e^3 . At stages s , where $\gamma \subseteq f_s$ we will put all balls $x \notin A_s$ such that $|\gamma| \leq x \leq s$ into $X_{\gamma_1, s}^3$. If $\gamma \subset f$ then almost all balls not in A are in $X_{\gamma_1}^3$. At the first stage where $\alpha(x, s) = \gamma$, we will put x into $X_{\gamma_2}^3$. If we remove x from the machine before entering A , we will put x into $X_{\gamma_3}^3$. Should f_s ever be to the left of γ , then some ball x with $\alpha(x, s) \subseteq f_s$ already in $X_{\gamma_1, s}^3$ might enter A without proper delay. However, since only finitely many such stages may occur along the true path whenever f_s moves to the left of γ , we may reset our construction of X_γ^3 (equivalently, we imagine that the tree guesses at how many elements each positive requirement places into A).

Given a stage $s + 1$ such that $\gamma \subseteq f_{s+1}$, let $t \leq s$ be the greatest stage such that $\gamma \subseteq f_t$ (if t does not exist let $t = 0$.) Define

$$l_\gamma(s) = \max x[(\forall z < x)\varphi_{e,s}(z) \downarrow].$$

The function $l_\gamma(s)$ measures the length of convergence of φ_e at stage s . If $l_\gamma(s) > l_\gamma(t)$ and, for all $x \in X_{\gamma_2, s}^3$, if $x \in X_{\gamma_2, \text{at } s'}$ then $l_\gamma(s) > s'$, then we say that $s + 1$ is γ -*expansionary*. In other words, stage $s + 1$ is γ -expansionary if the length of convergence of φ_e has increased and the proper amount of delay for all $x \in X_{\gamma_2, s}^3$ has been determined. At γ -expansionary stages $s + 1$, we move all balls x such that $\alpha(x, s) = \gamma$ downward so that $\alpha(x, s + 1) = \beta$,

where $\beta \hat{0} \subseteq \gamma$ and β is the greatest such subnode of γ assigned to some $\mathcal{N}_{e'}$ (only nodes working on the requirements $\mathcal{N}_{e'}$ stop balls from moving downwards) or if no such β exists let $\beta = \lambda$. If $s + 1$ is γ -expansionary, we let $\gamma \hat{0} \subset f_{s+1}$. Otherwise, we let $\gamma \hat{1} \subset f_s$. If we have moved any balls downwards or $|\gamma| = s$, we end this stage. Otherwise, we consider the action of $\gamma \hat{0}$ or $\gamma \hat{1}$.

\mathcal{N}_e is a Π_2^0 requirement. How \mathcal{N}_e is met depends on the answer to the Π_2^0 question is φ_e total. Suppose that $\gamma \subset f$. Define f such that $\gamma \hat{0} \subset f$ if φ_e is total and $\gamma \hat{1} \subset f$ if not. If $\gamma \subset f$ then it not hard to that $\liminf f_s \uparrow (|\gamma| + 1) = \gamma \hat{0}$ iff φ_e is total.

Note that we have made the simplifying assumption that if we enumerate x into $X_{\gamma_2}^3$ at stage s then $x \in X_{\gamma_2, s}^3$. While we may simply choose an enumeration of $X_{\gamma_2}^3$ to make this true, we must satisfy \mathcal{N}_e with respect to the canonical enumeration of elements into c.e. sets. However, using the recursion theorem, we may safely assume that each node is actually in possession of an index for every c.e. set built at that node and then, when necessary, we can simply wait until every element enumerated into some $X_{\gamma_2}^3$ appears in it in the canonical enumeration. Since such modifications are straightforward but tedious, we will refrain from further mention of them.

We assume the nodes that place balls on the machine obey the following rules and assumptions. A node $\alpha \supset \gamma$ can only place a ball x on the machine at stage t if $x \in X_{\gamma_1, t}^3$. Moreover, while α might place a ball on the machine at stage s , α can only place these balls at nodes working on the requirement $\mathcal{N}_{e'}$ for some e' . While we will not restrict how many balls α can place on the machine, we assume

\mathcal{A} Only finitely many balls that α places on the machine enter A .

Assume that $\gamma \hat{0} \subset f$. Let s' be such that for all $s \geq s'$, $\gamma \subseteq f_{s'}$, $f_s \not\leq_L \gamma$ and no $\alpha \subset \gamma$ places any more balls into A after stage s' . Under our extra Assumption \mathcal{A} , we know such a stage exists. Assume we are dealing with stages $s \geq s'$. It is not difficult to verify by induction on the length of γ that if $\gamma \hat{0} \subseteq f_s$, $\alpha(x, s - 1) = \gamma$, and $t > s$ is the next stage such that $\gamma \hat{0} \subseteq f_t$ then either $x \in A_t$ or $x \in X_{\gamma_3, t}^3$ and for all $y > s'$, if $\alpha(y, t) \subset \gamma$ then $\alpha(y, t) \hat{1} \subset \gamma$. It is easy to determine which balls enter A between such stages. We assumed that all balls placed on the machine by nodes $\alpha \subset \gamma$ that enter A have already entered by stage s' . Therefore, the balls that enter A between s' and t come from nodes to the right of γ . Since these nodes

were reset at stage s' , these balls all have to be larger than s' (otherwise we have that $\alpha(y, t') \leq_L \gamma$ for some stage $s' < t' < t$) and get into A by stage t . Hence, with the above movement of balls and Assumption \mathcal{A} we have that $X_\gamma^3 =^* \bar{A}$ and we have met \mathcal{N}_e .

Our next goal is to make A so that it is not computed by any 2-tardy. We must meet the requirements:

\mathcal{P}_e : If $\Phi_{e_1}(W_{e_2}) = A$, then W_{e_2} is not 2-tardy.

We will assign a parent node α to \mathcal{P}_e . Node α will be working on the requirement:

\mathcal{P}_α : If $\Phi_\alpha(W_\alpha) = A$ then W_α is not 2-tardy.

Determining whether $\Phi_\alpha(W_\alpha) = A$ is Π_2^0 . So α will have two outcomes 1 and 2: outcome 1 if $\Phi_\alpha(W_\alpha) = A$ and outcome 2 otherwise. We will later use outcome 0 to denote a Σ_1^0 win. Like above, determining whether $\Phi_\alpha(W_\alpha) = A$ can be measured by asking if there are infinitely many expansionary stages where length here measures length of agreement between $\Phi_\alpha(W_\alpha)$ and A .

Assume $\alpha \subseteq f_s$. Let $t \leq s$ be the greatest stage such that $\alpha \subseteq f_t$ (if t does not exist let $t = 0$.) Define $l_\alpha(s) = \max x[(\forall z < x)\Phi_{\alpha, s}^{W_\alpha}(z) \downarrow = A_s(z)]$. We say that $s + 1$ is γ -expansionary if

1. $l_\alpha(s) > l_\alpha(t)$ and,
2. for all $\beta \supseteq \alpha$, if x_β is defined (these will be witnesses to help meet requirement \mathcal{P}) then $l_\alpha(s) > x_\beta$.

If $s + 1$ is α -expansionary, let $\alpha \hat{1} \subseteq f_{s+1}$. Otherwise, $\alpha \hat{2} \subseteq f_{s+1}$. If there are only finitely many expansionary stages, we need not take any action to meet \mathcal{P}_α . We only need to take action if it appears there are infinitely many expansionary stages (the Π_2^0 outcome).

We can define the function $p_\alpha(t) = s$ iff $s > t$ is the least stage such that $\alpha \hat{1} \subset f_s$. If $\alpha \hat{1} \subset f$ then p_α is computable. From our work above, we know if $\alpha(x, t) \hat{0} \subset \alpha$ then at stage $s = p_\alpha(t)$ either x is in A or removed from the machine. This is the function we will try to use to witness W_α is not 2-tardy.

As a first approximation to showing W_α is not 2-tardy, we might try the following. Above the node $\alpha \hat{1}$, we will have nodes β working on the requirements:

$\mathcal{P}_{\alpha, \epsilon}$: If $X_e^2 = \bar{W}_\alpha$ then
 there exists y and s such that $y \in X_{e, s}^2$ and $y \in W_{\alpha, p_\alpha(s)}$.

The idea to meet $\mathcal{P}_{\alpha,e}$ is the following: At a stage s where $\beta \subset f_s$, choose some large ball x_β . Keep x_β out of A and off the machine. Let $u_{\alpha,s}(x)$ be the use of $\Phi_{\alpha,s}^{W_\alpha}(x)$. Wait for a stage s where $\alpha \hat{1} \subseteq f_s$, $X_{e,s}^2 \upharpoonright u_{\alpha,s}(x_\beta) = \overline{W}_{\alpha,s} \upharpoonright u_{\alpha,s}(x_\beta)$ and $\beta \subset f_s$. If such a stage can be found, we want to add x_β to A quickly, before stage $t = p_\alpha(s)$. Since $\Phi_\alpha^{W_\alpha} = A$, some ball $y < u_{\alpha,s}(x_\beta)$ must enter W_α by stage t . That y must be in $X_{e,s}^2$.

The problem is adding these balls into A quickly. If we could place the balls x_β that we want to enter A into the machine at some node $\gamma \subseteq \alpha$ at stage s , then by our work above we would not have a problem. Since there might be infinitely many $\mathcal{P}_{\alpha,e}$ that want to place balls into A , this would violate our extra Assumption \mathcal{A} . We might try to remove this extra assumption. But even so, the set of balls that all requirements $\mathcal{P}_{\alpha,e}$ might want to add to A is not computable. So, we have no reasonable way to manage these balls if we allow them all to enter the machine at α or below.

Hence, for each e we must assign a different node $\beta \supseteq \alpha \hat{1}$ to $\mathcal{P}_{\alpha,e}$. When β wants to add x_β to A , the node β places x_β at the largest substring $\gamma = \nu \hat{0}$ of β where ν is assigned to some $\mathcal{N}_{e'}$. Let stage t' be the first stage that x_β goes below α in the machine. At such a stage we have that $\alpha \hat{1} \subset f_{t'}$. If $X_{e,t'}^2 \upharpoonright u_{\alpha,t'}(x_\beta) = \overline{W}_{\alpha,t'} \upharpoonright u_{\alpha,t'}(x_\beta)$, we let x_β continue downwards into A for a win (on the above y and t') on $\mathcal{P}_{\alpha,e}$ as described above. But this may no longer be the case. We have no reason to believe that t' is expansionary for $X_e^2 = \overline{W}_\alpha$. It may be the case that at stage t' , $X_{e,t'}^2$ is already correctly predicting which balls y will enter W_α .

Hence, we must modify our requirements to

$$\mathcal{P}_{\alpha,e}: \quad \text{If } X_e^2 = \overline{W}_\alpha \text{ and } \neg[(\exists y)(\exists s)[y \in X_{e,s}^2 \wedge y \in W_{\alpha,p_\alpha(s)}]]$$

then for all i

$$\mathcal{P}_{\alpha,e,i}: \quad \text{If } X_i^2 = X_e^2 \text{ then } (\exists y)(\exists s)[y \in X_{i,s}^2 \wedge y \in W_{\alpha,p_\beta(s)}].$$

As before some $\beta \supset \alpha \hat{1}$ will be assigned to $\mathcal{P}_{\alpha,e}$. The node β will have three possible outcomes. The first, $\beta \hat{0}$, is in the case we have a Σ_1^0 win for $\mathcal{P}_{\alpha,e}$, i.e., a ball y and stage s where $y \in X_{e,s}^2$ and $y \in W_{\alpha,p_\alpha(s)}$. The second outcome $\beta \hat{1}$ holds if there is not a Σ_1^0 win and $X_e^2 = \overline{W}_\alpha$. The $\beta \hat{2}$ outcome holds otherwise. As above, we will measure whether $X_e^2 = \overline{W}_\alpha$ by expansionary stages.

Assume $\beta \subseteq f_{s+1}$. Let $t \leq s$ be the greatest stage such that $\beta \hat{1} \subseteq f_t$ (if t does not exist, let $t = 0$). Define $l_\beta(s) = \max x[(\forall z < x)[X_{e,s}^2(z) = \overline{W}_{\alpha,s}(z)]]$. We say that $s + 1$ is β -*expansionary* if

1. $l_\beta(s) > l_\beta(t)$ and
2. for all $\delta \supseteq \beta$, if x_δ (a ball to satisfy $\mathcal{P}_{\alpha,e,i}$) is defined then $l_\beta(s) > u_{e,s}(x_\delta)$.

If stage $s + 1$ is γ -expansionary and we have not seen a Σ_1^0 win for $\mathcal{P}_{\alpha,e}$, then $\beta \hat{1} \subseteq f_{s+1}$. If we have seen the Σ_1^0 win, then $\beta \hat{0} \subseteq f_{s+1}$. Otherwise, $\beta \hat{2} \subseteq f_{s+1}$.

If there are only finitely many expansionary stages or we see the Σ_1^0 win, $\mathcal{P}_{\alpha,e}$ is automatically satisfied. Assume that this is not the case. Hence, as above, we are in Π_2^0 outcome. In this case, we must meet $\mathcal{P}_{\alpha,e,i}$, for all i . For each i we will assign some node $\delta \supseteq \beta \hat{1}$ to $\mathcal{P}_{\alpha,e,i}$. The outcomes and approximations to the true path for δ are defined in similar fashion to what was done for β and we will not repeat them. The issue for δ is showing that δ does not have the Π_2^0 outcome, $\delta \hat{1}$.

At a stage s where $\delta \subset f_s$, choose a large unused ball x_δ , which we hold out of A and the machine. Wait for a stage s where $\delta \hat{1} \subseteq f_s$. If such a stage does not exist we have won this requirement. If such a stage exists, then place x_δ into the machine at the largest substring $\gamma = \nu \hat{0}$ of β (note, not δ) where ν is assigned to some $\mathcal{N}_{e'}$ and end this stage.

Now, assuming $\beta \hat{1} \subset f$, there will be a later stage t' where x_δ moves below α and $\alpha \hat{2} \subseteq f_{t'}$. Otherwise, $\alpha \hat{2} \subset f$ and then the action of α, β and γ are finitary and therefore Assumption \mathcal{A} holds. If $X_{e,t'}^2 \upharpoonright u_{e,t'}(x_\delta) = \overline{W}_{\alpha,t'} \upharpoonright u_{e,t'}(x_\delta)$, we let x_δ continue downwards into A for a win on $\mathcal{P}_{\alpha,e}$ as before. If this happens for any i , it will provide us with a Σ_1^0 win on $\mathcal{P}_{\alpha,e}$ and all the balls x_δ will be removed from the tree since they are to the right of the true path. Therefore, the action of β and those $\delta \supset \beta \hat{1}$ will be finitary. Hence, in this case, Assumption \mathcal{A} holds.

Assume that $X_{e,t'}^2 \upharpoonright u_{e,t'}(x_\delta) \neq \overline{W}_{\alpha,t'} \upharpoonright u_{e,t'}(x_\delta)$. Here, we will remove x_δ from the machine. We put x_δ into $X_{\gamma,3}^3$ at stage t' , for all γ such that x_δ is in $X_{\gamma,2}^3$ at stage t' . We now have to argue that this provides us with a win for $\mathcal{P}_{\alpha,e,i}$.

Remark 2.2. Assume that $\beta \hat{1} \subset f$. Since $p_\alpha(s) \geq s$, if it is ever the case that $X_{e,s}^2$ is a proper superset of $\overline{W}_{\alpha,s}$ then we know some ball y in $X_{e,s}^2 \cap W_{\alpha,s}$ must later leave $X_{e,s}^2$. Such a ball and a stage will provide us with a Σ_1^0 win for β . So, we can assume that \overline{X}_e^2 is a faster enumeration to W_α than the standard enumeration.

We wait for the next stage $t = p_\beta(s)$ such that $\beta^{\wedge}1 \subseteq f_t$. If such a stage does not exist, then $\beta^{\wedge}2 \subset f$, the action of β and all the related δ are finitary, and therefore, Assumption \mathcal{A} holds.

At this point we have the following

$$\begin{aligned} X_{i,s}^2 \upharpoonright u_{e,s}(x_\delta) &= X_{e,s}^2 \upharpoonright u_{e,s}(x_\delta) = \overline{W}_{\alpha,s} \upharpoonright u_{e,s}(x_\delta) \\ X_{e,t'}^2 \upharpoonright u_{e,t'}(x_\delta) &\neq \overline{W}_{\alpha,t'} \upharpoonright u_{e,t'}(x_\delta) \\ X_{e,t}^2 \upharpoonright u_{e,t}(x_\delta) &= \overline{W}_{\alpha,t} \upharpoonright u_{e,t}(x_\delta) \end{aligned}$$

If $\overline{W}_{\alpha,s} \upharpoonright u_{e,s}(x_\delta) \neq \overline{W}_{\alpha,t'} \upharpoonright u_{e,t'}(x_\delta)$, then some y entered $W_{\alpha,t'}$ after stage s . Then, we have a Σ_1^0 win for δ since $t' < p_\beta(s)$ and $y \in X_{i,s}^2$. So, assume that $\overline{W}_{\alpha,s} \upharpoonright u_{e,s}(x_\delta) = \overline{W}_{\alpha,t'} \upharpoonright u_{e,t'}(x_\delta)$. By the fact that $\beta^{\wedge}1 \subset f$ and Remark 2.2, it must be the case that

$$X_{e,t'}^2 \upharpoonright u_{e,t'}(x_\delta) \subsetneq X_{e,s}^2 \upharpoonright u_{e,s}(x_\delta).$$

Hence, some ball $y < u_{e,s}(x_\delta)$ must leave $X_{e,s}^2$. Since X_e^2 is 2-c.e. that ball y can never return. Hence, since $\beta^{\wedge}1 \subset f$, that ball y must enter W_α and, moreover, it must enter before stage $t = p_\beta(s)$. Therefore, we have a Σ_1^0 win for δ .

Assume that $\beta^{\wedge}1 \subset f$. The infinitely many δ above $\beta^{\wedge}1$ might place infinitely many balls onto the machine. Moreover, we can arrange things such that the set of these balls is not a c.e. set. But at most one of these balls will enter A and Assumption \mathcal{A} holds.

All that remains at this point is to assign the nodes on the tree such that all the requirements are met. But this can be done in a straightforward fashion. \square

3. Definability and n -tardies

We define a property Q_n such that Q_n is nontrivial and if $Q_n(A)$ holds, then A is n -tardy. In the next section, we define a nontrivial property \hat{Q}_n using Q_n such that if $\hat{Q}_n(A)$ holds, then A is n -tardy and $\neg\hat{Q}_i(A)$ holds for all $i < n$.

Definition 3.1. (i) Let $A \subset_\infty C$ denote that $A \subseteq C$ and $C - A$ is infinite.
(ii) A subset A is a *major subset* of C , denoted $A \subset_m C$ if $A \subset_\infty C$ and for all e ,

$$\bar{C} \subseteq W_e \implies \bar{A} \subseteq^* W_e.$$

(If $A \subset_m C$, then A and C are noncomputable.)

(iii) $A \sqsubset C$ if there exists a B such that $A \sqcup B = C$, i.e., $A \cup B = C$ and $A \cap B = \emptyset$.

3.1. Q_n

We begin by defining Q_n for $n \geq 2$ and proving that if $Q_n(A)$ holds, then A is n -tardy. The definition of Q_n generalizes Definition 3.2 of Q_2 given in [6]. We define Q_n separately for n even and odd.

Definition 3.2.

$$\begin{array}{l}
Q_{2n}(A) \quad (\exists C \supset_m A) \\
\quad (\forall B_1 \subseteq C)(\forall B_2 \subseteq B_1) \dots (\forall B_n \subseteq B_{n-1}) \\
\quad (\exists D_1 \subseteq C)(\exists D_2 \subseteq D_1) \dots (\exists D_n \subseteq D_{n-1}) \\
\quad (\forall S \sqsubset C)(\exists T_1 \supseteq \bar{C})(\exists T_2 \subseteq T_1) \dots (\exists T_n \subseteq T_{n-1}) \\
\\
Q_{2n}(A) : \text{if} \quad \left[\begin{array}{l} B_1 \cap (S - A) = D_1 \cap (S - A) \\ B_2 \cap (S - A) = D_2 \cap (S - A) \\ \dots \\ B_n \cap (S - A) = D_n \cap (S - A) \end{array} \right] \\
\\
\quad \implies \\
\\
Q_{2n}(A) : \text{then} \quad \left[\begin{array}{l} (A \cup T_2) \cap (S \cap T_1) = B_1 \cap (S \cap T_1) \\ (A \cup T_3) \cap (S \cap T_2) = B_2 \cap (S \cap T_2) \\ \dots \\ (A \cup T_n) \cap (S \cap T_{n-1}) = B_{n-1} \cap (S \cap T_{n-1}) \\ A \cap (S \cap T_n) = B_n \cap (S \cap T_n) \end{array} \right]
\end{array}$$

$$Q_{2n+1}(A) \quad (\exists Y \subseteq \bar{A})Q_{2n}(A \cup Y)$$

Theorem 3.3. *If $Q_n(A)$ holds, then A is n -tardy.*

We break this proof into two lemmas, one handling the case where n is even and the other handling the case where n is odd.

Lemma 3.4. *If $Q_{2n}(A)$ implies A is $2n$ -tardy for any c.e. set A , then $Q_{2n+1}(A)$ implies A is $2n + 1$ -tardy for any c.e. set A .*

Proof. If $Q_{2n+1}(A)$ then $Q_{2n}(A \cup Y)$ holds for some Y disjoint from A . By assumption $A \cup Y$ is $2n$ -tardy. Thus, if $p(s)$ is a total computable function, there is some $2n$ -c.e. set $X^{2n} = X_1 - X_2 + X_3 - \dots - X_{2n}$ equal to $\overline{A \cup Y}$ such that $x \in X_s^{2n} \implies x \notin A_{p(s)}$. Let

$$X^{2n+1} = X^{2n} + Y = X_1 - X_2 + X_3 - \dots - X_{2n} + Y.$$

Since $Y \cap A = \emptyset$, $X^{2n+1} = \bar{A}$ and $x \in X_s^{2n+1} \implies x \notin A_{p(s)}$. Since $p(s)$ was arbitrary, A is $2n + 1$ -tardy. \square

Lemma 3.5. *$Q_{2n}(A)$ implies A is $2n$ -tardy.*

This is a generalization of Theorem 3.3 in [6] and we retain their approach but present it as a modern Π_2^0 guessing argument.

Proof. Fix A and C (and indexes for them) such that A satisfies $Q_{2n}(A)$ via C and $A \subseteq C \searrow A$ where the latter property can be guaranteed purely by change of index. Following [6], we think of $Q_{2n}(A)$ as a two player game between the \exists -player (called EXISTS) who plays the sets $\vec{D} = (D_1, D_2, \dots, D_n)$ and $\vec{T} = (T_1, \dots, T_n)$ and the \forall -player (called FORALL) who plays the sets B_1, B_2, \dots, B_n and $S \sqsubset C$. Should $A, C, S, \vec{D}, \vec{T}$ witness the satisfaction of $Q_{2n}(A)$ we say the EXISTS player wins. Otherwise, the FORALL player wins. Since C witnesses the satisfaction of $Q_{2n}(A)$, the EXISTS player must have a winning strategy. Given any total computable function $p(s)$, the proof will proceed by specifying a strategy for the FORALL player such that winning response \vec{D}, \vec{T} of the EXISTS player allows us to build a $2n$ -c.e. set X^{2n} witnessing that A is $2n$ -tardy.

Given a total computable function $p(s)$, FORALL will respond by building \vec{B} . However, in the construction of B , FORALL will want to use information about the particular sets \vec{D}, \vec{T} played by EXISTS, but \vec{B} must be built without knowledge of \vec{D} or \vec{T} . We let \vec{B} react to the particular choice of \vec{D} by simultaneously building \vec{B} and a sequence of sets $S_e \sqsubset C$ such that on S_e , the collection \vec{B} plays against $\vec{D}_e = (W_{e_0}, W_{e_1}, \dots, W_{e_n})$. During this construction, \vec{B} will be built so that, for every e , property $(Q_{2n}(A) : \text{if})$ holds for $S = S_e, \vec{D} = \vec{D}_e$. Thus, for EXISTS to have a winning strategy, there must be some \vec{T} witnessing the satisfaction of $(Q_{2n}(A) : \text{then})$.

We now further divide up the sets S_e into the sets $S_{e,j}$ with $S_e = \bigsqcup_{j \in \omega} S_{e,j}$ so that FORALL builds \vec{B} to play against \vec{D}_e, \vec{T}_j on $S_{e,j}$. Since S must be played without knowledge of \vec{T} , we appear to run the risk that the winning strategy for EXISTS never plays \vec{T}_j against $S_{e,j}$. However, since \vec{B}, \vec{D}_e, S_e satisfies $(Q_{2n}(A) : \text{if})$, there is some j such that \vec{B}, \vec{T}_j, S_e satisfy $(Q_{2n}(A) : \text{then})$. But as $S_{e,j} \subset S_e$, it follows that $\vec{B}, \vec{T}_j, S_{e,j}$ satisfy $(Q_{2n}(A) : \text{then})$. Thus, provided for all e we maintain $(Q_{2n}(A) : \text{if})$ for S_e, \vec{D}_e , we may assume that for some e, j the sets $\vec{B}, \vec{D}_e, \vec{T}_j, S_{e,j}$ satisfy both $(Q_{2n}(A) : \text{if})$ and $(Q_{2n}(A) : \text{then})$.

We let α range over indexes e, j for n tuples of c.e. sets and define

$$\begin{aligned} S_\alpha &= S_{e,j} \\ D_i^\alpha &= W_{e_i} \\ T_i^\alpha &= W_{j_i} \end{aligned}$$

where we stipulate that our indexes satisfy

$$\begin{aligned} D_1^\alpha &\subseteq C \\ D_{i+1}^\alpha &\subseteq D_i^\alpha \\ T_{i+1}^\alpha &\subseteq T_i^\alpha \end{aligned}$$

Relative to a particular choice of \vec{B} , the predicate $F(\alpha)$ asserting that the sets $\vec{B}, \vec{D}_e, \vec{T}_j, S_{e,j}$ satisfy both $(Q_{2n}(A) : \text{if})$ and $(Q_{2n}(A) : \text{then})$ is Π_2^0 . Thus, there is a uniformly computable sequence of predicates $F_s(\alpha)$ referring only to the commitments we have made about \vec{B} by stage s in our construction such that $F(\alpha) \leftrightarrow (\exists^\infty s) F_s(\alpha)$. Using this predicate, we define a strong array of finite sets U_k^α for every α and $k \in [1, n]$ as follows.

$$\begin{aligned} x \in U_{1,s}^\alpha &\iff x \in U_{1,s-1}^\alpha \vee [s \geq x \wedge x \in (T_{1,s}^\alpha - C_s) \wedge F_s(\alpha)] \\ x \in U_{l,s}^\alpha &\iff x \in U_{l-1,s}^\alpha \cap T_{l,s}^\alpha \end{aligned}$$

By way of the slowdown lemma [9] applied to the above arrays, we define

$$X_{2i-1}^\alpha = \bigcup_{s \in \omega} U_{i,s}^\alpha$$

satisfying

$$X_{2i-1,s}^\alpha \subset U_{i,s}^\alpha$$

If we build $S_{e,j}$ as described, there must be some least α for which $F(\alpha)$ holds by the remarks above. For that α , we have $X_1^\alpha \supset^* \bar{A}$ since $(Q_{2n}(A))$ requires that $T_1 \supset \bar{C}$ and $F_s(\alpha)$ holds for infinitely many s . So, $U_1^\alpha \supset \bar{C}$ and $A \subset_m C$. We also have $T_{i+1} \subseteq T_i$ and $X_{2i-1}^\alpha = T_i^\alpha \cap X_1^\alpha$ by definition. So, if the sequence \vec{T}^α witnesses that $(Q_{2n}(A) : \text{then})$ holds, we may replace each T_i^α with X_{2i-1}^α without falsifying $(Q_{2n}(A) : \text{then})$.

We now build S_α with the intention that (with finitely many exceptions) every element that is in $X_1^\alpha \cap A$ is in S_α . If $x \in C_{s+1} - C_s$, take the least α such that $x \in U_{1,s}^\alpha$ and enumerate x into S_α . If no such α exists, enumerate x into the garbage set S_{-1} . Note that $C = \bigsqcup_{\alpha \in 2^{<\omega}} S_\alpha \sqcup S_{-1}$ by construction, so, $S_\alpha \sqsubset C$ for every α . Furthermore, by construction, once x enters C it can no longer enter U_1^α for any α . Suppose α is the least such that $F(\alpha)$ holds. Since U_1^β is finite for every $\beta < \alpha$, we have $U_1^\alpha \cap C \subset^* S_\alpha$. Hence, for all $i \in [1, n]$

$$(3.6) \quad X_{2i-1}^\alpha \cap C \subset^* S_\alpha.$$

Conversely, $S_\alpha \subseteq X_1^\alpha$. We are now ready to define \vec{B} and the even components of X^α . Let

$$X_{2i}^\alpha = S_\alpha \cap D_i^\alpha$$

where by way of the slowdown lemma [9], we ensure that

$$X_{2i+2}^\alpha \subseteq X_{2i}^\alpha \searrow X_{2i+2}^\alpha$$

Since $X_1^\alpha \cap C \subset^* S_\alpha$ and $X_1^\alpha \cap \bar{C} \subseteq \bar{A}$, requiring X_{2i}^α to be a subset of S_α is no handicap to ensuring $X^\alpha = \bar{A}$. If $F(\alpha)$ holds, then we claim that

$$(3.7) \quad \begin{aligned} X_{2j-1}^\alpha \cap X_{2j}^\alpha \cap \bar{A} &\subseteq X_{2j+1}^\alpha \\ X_{2n-1}^\alpha \cap X_{2n}^\alpha \cap \bar{A} &= \emptyset. \end{aligned}$$

To see this, let $x \in X_{2j-1}^\alpha \cap X_{2j}^\alpha \cap \bar{A}$. As $X_{2j}^\alpha = D_j^\alpha \cap S_\alpha$, we have $x \in D_j^\alpha \cap (S_\alpha - A)$ which by $(Q_{2n}(A) : \text{if})$ is contained in B_j . By a prior remark, we may substitute X_{2j-1}^α in for T_j in $(Q_{2n}(A) : \text{then})$ and as $x \in B_j \cap S_\alpha \cap X_{2j-1}^\alpha$, we have $x \in A \cup X_{2j+1}^\alpha$. Since $x \notin A$, we have $x \in X_{2j+1}^\alpha$. Moreover, by similar reasoning, $X_{2n-1}^\alpha \cap X_{2n}^\alpha \cap \bar{A} = \emptyset$. We then derive the following containments.

$$(3.8) \quad \begin{aligned} \bar{A} \cap S_\alpha &\subseteq X^\alpha \\ \bar{A} &\subseteq^* X^\alpha \end{aligned}$$

For the first containment, if $x \in \bar{A} \cap S_\alpha$ then, as $S_\alpha \subseteq X_1^\alpha$, there is a maximal j such that $x \in X_{2j-1}^\alpha$. Since the even indexed components of X^α are nested, if $x \notin X_{2j}^\alpha$, then $x \in X^\alpha$, and we are done. If $x \in X_{2j}^\alpha$, then (3.7) yields a contradiction. The second containment follows since $X^\alpha \supseteq X_1^\alpha - S_\alpha$ (by definition, each $X_{2j}^\alpha \subseteq S_\alpha$, so no elements outside of S_α are removed from X^α) and $X_1^\alpha \supseteq^* \bar{A}$. We now define \bar{B} to that the other direction of containment and the tardiness property hold.

$$(3.9) \quad x \in B_i \iff (\exists \alpha)(\exists s) [x \in X_{2i,s}^\alpha \wedge x \notin A_{p(s)}]$$

Tracing out the definition of X_{2i}^α , it is evident that on $S_\alpha - A$ we have $B_i = D_i^\alpha = W_{e_i}$. Hence, by our earlier arguments, there is some α such that $F(\alpha)$ holds. Now let α be the least such. Since $B_i \cap S_\alpha \subseteq D_i^\alpha \cap S_\alpha$, using $(Q_{2n}(A) : \text{then})$ we see

$$A \cap X_{2i-1}^\alpha \cap S_\alpha \subseteq B_i \cap X_{2i-1}^\alpha \cap S_\alpha \subseteq D_i^\alpha \cap S_\alpha = X_{2i}^\alpha.$$

Thus, if $x \in A \cap S_\alpha$ then $x \notin X^\alpha$. By (3.6), $X^\alpha \cap C \subseteq^* S_\alpha$. Since $X^\alpha \cap C \subseteq^* S_\alpha$ and $A \subseteq C$, this entails $\bar{A} \supseteq^* X^\alpha$. Putting this together with (3.8), we conclude

$$\begin{aligned} \bar{A} \cap S_\alpha &= X^\alpha \cap S_\alpha \\ \bar{A} &=^* X^\alpha \end{aligned}$$

We now argue that X^α has the desired tardiness properties. Suppose $x \in X_1^\alpha$ and $x \in A \cap S_\alpha$. Let j be the greatest such that $x \in X_{2j-1}^\alpha$. Now suppose x enters X_{2j}^α at stage s . If $x \in A_{p(s)}$ then by (3.9) $x \notin B_i$. But as $x \in X_{2j-1}^\alpha \cap S_\alpha$, it follows from $(Q_{2n}(A) : \text{then})$ that $x \notin A$. This is a contradiction. Therefore,

$$x \in S_\alpha \cap X_s^\alpha \implies x \notin A_{p(s)}$$

Now set $X_{2i} = X_{2i}^\alpha$ and $X_{2i-1} =^* X_{2i-1}^\alpha$ where we build X_{2i-1} by removing the finitely many members by (3.6) of $A \cap \bar{S}_\alpha \cap X_1^\alpha$ from X_{2i-1}^α and adding the

finitely many members of $\bar{A} - X_1^\alpha$. The set X witnesses that A is $2n$ -tardy with respect to $p(s)$. Since $p(s)$ was arbitrary, we can conclude A is $2n$ -tardy. \square

Taken together these lemmas suffice to establish Theorem 3.3.

4. Proper Satisfaction Q_n

At this point we have a countable collection of properties Q_n for $n \geq 2$ preserved under automorphism guaranteeing incompleteness. It is easily verified that $Q_n(A)$ implies $Q_{n+1}(A)$ so to illustrate countably many incomplete orbits, we must show this hierarchy of properties does not collapse. In particular, it will suffice to show that for every $n > 2$ there is a properly n -tardy A satisfying $Q_n(A)$ as we can then define

$$\hat{Q}_n(A) \iff Q_n(A) \wedge \neg Q_{n-1}(A) \wedge \dots \wedge \neg Q_2(A)$$

Since by Theorem 3.3, any set satisfying $Q_n(A)$ must be n -tardy it follows that our set A satisfies $\hat{Q}_n(A)$ and that the properties \hat{Q}_n for $n \geq 2$ give countably many disjoint orbits.

Theorem 4.1. *For all $m \geq 2$ there is a properly m -tardy A satisfying $Q_m(A)$.*

Again we consider the even and odd cases separately. We first work to show that there is a properly $2n$ -tardy satisfying Q_{2n} and then modify this argument to yield a properly $2n + 1$ -tardy satisfying Q_{2n} .

Ideally our argument in the even case would establish that any $2n$ -tardy that has a major superset satisfies Q_{2n} , but this appears to be insufficient. The role played by C is not only that of a major superset but also provides an early warning that something in T_1 may threaten to enter A (i.e., enter T_2). That is, we must wait until elements enter C before we can target them for entry into A . Thus, we first construct sets A and C with these properties.

4.1. Building A and C

Lemma 4.2. *For every $n \geq 1$ there is a properly $2n$ -tardy set A and a c.e. set C with $C \supset_m A$ such that for every total computable function p there is*

a $2n$ -c.e. set X_e^{2n} such that

$$(4.3a) \quad \bar{A} = X_e^{2n} = (X_{e_1}^{2n} - X_{e_2}^{2n}) \cup \dots \cup (X_{e_{2n-1}}^{2n} - X_{e_{2n}}^{2n})$$

$$(4.3b) \quad (\forall x, s) (x \in X_{e,s}^{2n} \implies x \notin A_{p(s)})$$

$$(4.3c) \quad (\forall k < 2n) [X_{e_{k+1}}^{2n} = C \searrow X_{e_{k+1}}^{2n}]$$

$$(4.3d) \quad X_{e_1}^{2n} \supseteq X_{e_2}^{2n} \supseteq \dots \supseteq X_{e_{2n-1}}^{2n} \supseteq X_{e_{2n}}^{2n}$$

$$(4.3e) \quad (\forall i < 2n) [X_{e_{i+1}}^{2n} = X_{e_i}^{2n} \searrow X_{e_{i+1}}^{2n}]$$

Proof of Lemma 4.2. To prove the claim, we start with a simple set \hat{C} with the property that $|\hat{C} \upharpoonright_{2x}| \leq x$ and simultaneously construct $C \supseteq \hat{C}$ and A . During this construction, we refer to the index of C as a c.e. set so we can measure its speed of enumeration. We justify this circularity by regarding the construction as a computable function operating on a guess at the index for C and returning an index for the resulting C we build and then applying the recursion theorem. To effect the construction of A , we will work to meet the requirements $\mathcal{N}_e, \mathcal{M}_e, \mathcal{R}_e$ specified below to which we assign priorities $3e, 3e + 1, 3e + 2$, respectively. These requirements are thought of as being laid out vertically in order of priority. Ultimately, the true construction will take the form of a tree argument in the style of 2.1. Rather than repeat the standard details of the tree layout, we instead present the argument as if it were a infinite injury pinball argument so as not to hide the real work in the magic of the tree machinery. Ultimately, however, we will observe that the computable corrections required by infinite injury can simply be considered as the action of the tree when phrased as a Π_2^0 tree argument and can thus be squared with requirements (4.3d) and (4.3e). Thus the construction may be regarded as a pinball machine with A at the bottom of the machine and the requirements stretching upwards.

During the construction, balls (numbers) will be released at requirements of the form \mathcal{M}_e and \mathcal{R}_e . These balls attempt to flow down through the negative requirements below. When (and if) they reach the bottom, they are enumerated into A . The organization of this construction is in principle similar to that used before but the extra complexity of a full Π_2^0 tree construction is unnecessary so we abandon it for clarity.

Each negative requirement \mathcal{N}_e will construct a $2n$ -c.e. set X_e^{2n} in the hope of meeting the requirement below, falling short only by virtue of computable injury. By pausing the construction until elements appear in the canonical enumeration of C , we may insist that (4.3c) holds.

$$\begin{array}{c}
(\exists x)(\varphi_e(x)\uparrow) \\
\mathcal{N}_e \qquad \qquad \text{or} \\
[X_e^{2n} = \overline{A}] \ \& \ (\forall x)(\forall s)[x \in X_{e,s}^{2n} \implies x \notin A_{\varphi_e(s)}]
\end{array}$$

We act to meet this requirement as follows. At the start of stage $s > e$, fix l to be maximal such that $(\forall x < l)[\varphi_{e,s}(x)\downarrow]$ and put every $x < l$ into X_{e_1} that is not already in A or located below \mathcal{N}_e along our list of requirements. If a ball x targeted for A by a weaker priority requirement reaches \mathcal{N}_e at stage s and it is not yet in X_{e_1} , it is immediately allowed to fall through to the next negative requirement along the path to A . Otherwise, if $x \in X_{e_1}$ let j be the largest index such that $x \in X_{e_j}$. Place x into $X_{e_{j+1}}$ and delay x from falling through to the next negative requirement until the first stage t such that $\varphi_{e,t}(s)\downarrow$ is reached. Whenever a weaker requirement decides to cancel its attempt to place some $x \in X_{e_{2j}} - X_{e_{2j+1}}$ into A before completion, x is placed into $X_{e_{2j+1}}$. Note that if φ_e is partial X_{e_1} will be finite and only finitely many balls will be permanently delayed by \mathcal{N}_e .

This construction suffices to meet \mathcal{N}_e modulo the balls put into A by higher priority requirements. At the end of the construction, we will observe that the set of elements that so slip past \mathcal{N}_e is computable. Thus, we can modify X_e^{2n} to satisfy the requirement without sacrificing any of the desired properties. We now work to ensure that $C \supseteq_m A$ via the following requirement.

$$\mathcal{M}_e \qquad \qquad W_e \supseteq \overline{C} \implies W_e \supseteq^* \overline{A}$$

Given C that we are building, $A \subset_m C$ requires that if $W_e \supset \overline{C}$ then $C - A \subseteq^* W_e$. If we knew from the outset that $W_e \cap C$ was infinite and $W_e \supseteq \overline{C}$, we would ensure all but finitely many members of $C - A$ are in W_e by enumerating elements into A . Fixing C to be a simple set ensures that if $W_e \supseteq \overline{C}$ then $W_e \cap C$ is infinite. Thus, taking C simple would suffice to satisfy \mathcal{M}_e . Since we cannot determine whether W_e is infinite, we instead assume that we have seen the entirety of W_e and correct our construction if we see more elements enter W_e . In particular, if we see W_e extend to contain $\overline{C} \cap [0, l]$, we can respond by enumerating (almost) every $x < l$ with $x \in C$ into A to keep $C - A \subseteq^* W_e$. To ensure $C - A$ is infinite, we absolve the first e (candidate) members of $C - A$ from being affected by \mathcal{M}_e .

We fix $\{C_s\}_{s \in \omega}$, a stage wise approximation to C , such that C_0 is some infinite computable subset of \hat{C} and other elements enter C_s only when they are enumerated into \hat{C} or placed into C by \mathcal{R}_e , which we describe below. We also fix a countable collection of markers m_k shared across all the \mathcal{M}_e requirements whose position at stage s we denote by $m_{k,s}$ with the intention (which we almost fulfill) of letting them come to rest on $C - A$. We describe the motion of these markers in terms of an e -state construction. Instead of maximizing the e -state of our markers, which would guarantee that any c.e. set containing infinitely many elements from $C - A$ contains almost all of them, we only maximize the e -state for c.e. sets threatening to contain \overline{C} . To this end, we employ the following twist on the notion of an e -state.

Definition 4.4. The C -complementing e -state of x , denoted $\epsilon_e^C(x)$, is defined to be the $\nu \in 2^{e-1}$ such that

$$(\forall i < e) (\nu(i) = 1 \leftrightarrow W_i \supseteq (\overline{C} \cap [0, x]) \wedge x \in W_i)$$

The C -complementing e -state of x at stage s , $\epsilon_e^C(x, s)$ is defined to be the $\nu \in 2^{e-1}$ such that

$$(\forall i < e) (\nu(i) = 1 \leftrightarrow W_{i,s} \supseteq (\overline{C}_s \cap [0, x]) \wedge x \in W_{i,s})$$

At the start of the construction, we place m_k on the k -th element of C_0 . At the start of stage $s + 1$ for every $x \in C_{s+1} - C_s$, we pick the least k such that $m_{k,s} > x$ and shift the markers after m_k down to their predecessors' location to fill the gap. Note that since $m_{k,s} > x$ if $\epsilon_e^C(m_{k,s}, s)(i) = 1$ then since $x \notin C_s$ it follows that $x \in W_{i,s}$ and thus $\epsilon_e^C(m_{k,s+1}, s+1) \geq_L \epsilon_e^C(m_{k,s}, s)$ where \geq_L denotes the lexicographic order.

After all the requirements with greater priority than \mathcal{M}_e have acted at stage s , we search for the least k, k' with $k' > k \geq e$ and

$$(4.5) \quad \epsilon_{e+1}^C(m_{k',s}, s) >_L \epsilon_{e+1}^C(m_{k,s}, s).$$

We then move the marker m_k to the location occupied by $m_{k'}$. We shift the later markers up accordingly and target the locations previously occupied by m_j for $k \leq j < k'$ for entry into A . Note that some of these elements may be reserved by higher priority requirements described later. In this case, we still move the markers but respect the reservation. Likewise, the motion of the markers is not affected if one of the elements targeted for A is permanently restrained by a negative requirement.

We inductively argue that each marker comes to rest. Pick s large enough so that every $m_{k'}$ for $k' < k$ has already come to rest on its final position and then select $s' > s$ to maximize $\epsilon_{k+1}^C(m_{k,s'}, s')$ for $s' \geq s$. The marker m_k cannot be moved at this point unless new elements are enumerated into C . By the above remarks, this movement cannot decrease $\epsilon_{k+1}^C(m_{k,s'}, s')$. Eventually, no further elements of C are enumerated below m_k , and the marker comes to rest. We let $m_{k,\infty}$ denote the location that m_k settles upon permanently.

We now argue that \mathcal{M}_e is satisfied if all but finitely many of the elements targeted for A by \mathcal{M}_e eventually enter A . To see this, fix some $W_e \supseteq \overline{C}$ and note that the intersection of all sets W_i for $i \leq e$ such that $W_i \supseteq \overline{C}$ contains \overline{C} . By the simplicity of C , this intersection must have an infinite intersection with C . It follows that all but finitely many elements of $C - A$ are contained in W_e and, indeed, all but finitely many elements in $C - A$ have some C -complementing $(e + 1)$ -state ν . Moreover, those elements targeted for A by \mathcal{M}_e form a computable set as, for large enough x , \mathcal{M}_e targets x for A only if it has done so by the time we see a marker above x attain the C -complementing e -state ν .

Lastly, we must also guarantee that A is not $(2n - 1)$ -tardy by meeting the requirements \mathcal{R}_e . To that end, we ensure every $(2n - 1)$ -c.e. set fails to provide a tardy approximation to \overline{A} . We fix a monotonically increasing computable function $p(s)$ such that if our construction directs us to place x into A at stage s then $B_{p(s)}$ for B a c.e. set given in the canonical enumeration. Provided $B = A$ then $p(s)$ will be total. By meeting the following requirements, we guarantee that no $(2n - 1)$ -c.e. approximation Y_e^{2n-1} to \overline{B} can work $p(s)$ steps ahead of B if $B = A$. Since the requirements dealing with $B = W_i$ do not interact significantly with those requirements working against $B = W_{i'}$, we drop the subscript i from the statement of the requirement.

$$\mathcal{R}_e \quad (\exists x) \left[\begin{array}{c} (\exists s)[x \in Y_{e,s}^{2n-1} \wedge x \in (A_s - A_{s-1})] \\ \vee \\ Y_e^{2n-1}(x) \neq \overline{A}(x) \vee A \neq B \end{array} \right]$$

So long as we believe \mathcal{R}_e to be unsatisfied, our strategy will work to provide \mathcal{R}_e another ball x it alone controls and let \mathcal{R}_e hold x out of A until we later see Y_e^{2n-1} change its mind to guess that $x \in \overline{B}$. At this point, we

target x for entry into A , and if x gets into A before it leaves Y_e^{2n-1} , we satisfy \mathcal{R}_e . If instead x leaves Y_e^{2n-1} before x enters A , we simply cancel our targeting of x for A , returning it back to wait at \mathcal{R}_e and again hold it out of A until the situation changes again. Since Y_e^{2n-1} can change its mind one less time than A can, Y_e^{2n-1} eventually must either fail to respond to these threats or x must enter the $(2n - 1)$ -th component of Y_e^{2n-1} . In the second case, we may safely put x into A since if Y_e^{2n-1} really witnessed the tardiness of B , we must have $x \notin B$. Note that we only place a new element into A after placing it in C if it is not already present there. The real complexity in meeting this requirement is guaranteeing that we will eventually reserve some x large enough that x is not permanently restrained by higher priority negative requirements and x has a large enough C -complementing e -state so that it is not co-opted by any higher priority \mathcal{M}_e .

During the construction, each \mathcal{R}_e will reserve a finite collection of intervals $\{[l_i^e, h_i^e]\}_{i=1}^{n_e}$ for its exclusive use such that if $e \neq e'$ or $i \neq i'$ then $[l_i^e, h_i^e]$ and $[l_{i'}^{e'}, h_{i'}^{e'}]$ are disjoint. Inside each interval, \mathcal{R}_e will maintain a marker r_i^e with position $r_{i,s}^e$ at stage s to indicate the currently active element in that interval. Only \mathcal{R}_e or a higher priority $\mathcal{M}_{e'}$ is allowed to target a member of $[l_i^e, h_i^e]$ for A . Whenever any marker r_i^e has been targeted for A but delayed by negative requirements, we act to reserve another interval for \mathcal{R}_e . If \mathcal{R}_e already has $j - 1$ intervals we select l_j^e to be the first element larger than every previously defined interval for any requirement. We then select h_j^e to be the least number currently occupied by some marker m_k such that $2(l_j^e + 2^e + 1) < h_j^e$ and place r_j^e on h_j^e . This has the effect of guaranteeing that there are at least 2^e elements in $[l_j^e, h_j^e]$ that will not be enumerated into \hat{C} , leaving \mathcal{R}_e complete control over placing these elements into C .

If at any point we observe that the first clause in (\mathcal{R}_e) has been satisfied, we mark \mathcal{R}_e satisfied and take no more action on its behalf. Otherwise, we act only if some r_j^e occupies x and either x is not currently targeted to enter A but $x \in Y_{e,s}^{2n-1}$ or x is currently targeted to enter A but $x \notin Y_{e,s}^{2n-1}$. In the former case, we target x for entry into A and in the latter case, we cancel our targeting of x for A (placing x into those X_e^{2n} being built below \mathcal{R}_e). If at some stage s , element $r_{i,s}^e$ is targeted for A by a higher priority $\mathcal{M}_{e'}$, then set $r_{i,s+1}^e$ to the largest $x < r_{i,s}^e$ (which must be in the reserved interval) with $x \notin C_s$ and enumerate x into C .

We argue that each \mathcal{R}_e only reserves finitely many elements and is eventually satisfied. Note that we only move r_i^e at stage s if some element $y > r_{i,s}^e$

increases its approximate C -complementing $(e + 1)$ -state above that of $r_{i,s}^e$. By enumerating $r_{i,s+1}^e$ into C , we cause the marker m_k occupying the least $y' > r_{i,s+1}^e$ to be shifted down to $r_{i,s+1}^e$. By our remark in the discussion of \mathcal{M}_e , we know that $\epsilon_{e+1}^C(r_{i,s+1}^e, s + 1) \geq_L \epsilon_{e+1}^C(y, s)$ when we move r_i^e , as the new location of r_i^e must have already entered any c.e. set contributing to $\epsilon_{e+1}^C(y, s)$. Combining these inequalities, we see that $\epsilon_{e+1}^C(r_{i,s+1}^e, s + 1) >_L \epsilon_{e+1}^C(r_{i,s}^e, s)$. As there are only 2^e many C -complementing $(e + 1)$ -states, we can move r_i^e at most $2^e - 1$ times. By choice of h_i^e , we know that each time we can find some element in $[l_i^e, h_i^e]$ not yet in C_s . Hence, r_i^e eventually occupies a location that is not stolen by a higher priority majorness requirement. Now, if \mathcal{R}_e only reserves finitely many intervals, it is satisfied, so assume it reserves infinitely many intervals. In this case, let $[l_i^e, h_i^e]$ be an interval with l_i^e so large that no element in this interval is permanently restrained by any $\mathcal{N}_{e'}$ for $e' \leq e$, and let x be the location r_i^e settles upon. But now x will eventually settle down into a victory against Y_e^{2n-1} as described above and no more intervals will be reserved for \mathcal{R}_e .

This completes the construction. We now need to verify that it has the claimed properties. If $\text{dom } \varphi_e$ is infinite then eventually every element not in A must settle down into X_e^{2n} or one of the finitely many negative requirements $\mathcal{N}_{e'}$ with $e' < e$ and $\text{dom } \varphi_e$ finite. Thus, after joining these finitely many trapped balls to the first odd component of X_e^{2n} they have not yet entered we can assume that X_e^{2n} contains \bar{A} . Moreover we can make this finite adjustment without disrupting the property that balls enter the earlier components of the $2n$ -c.e. set before the later ones and only enter $X_{e_2}^{2n}$ after C . Conversely, X_e^{2n} is contained in \bar{A} union the set of elements placed into A by requirements of the form $\mathcal{R}_{e'}$ or $\mathcal{M}_{e'}$ for $e' < e$ which in the former case is a finite set and the latter a computable set. Thus, X_e^{2n} is contained in \bar{A} union a computable subset K of A so we can fix X_e^{2n} to be equal to \bar{A} by simply intersecting \bar{K} with every positive (odd) component of X_e^{2n} . Since we do not alter the negative components of X_e^{2n} , we do not slow down any elements from leaving X_e^{2n} , and so retain the required tardiness property. But, by taking elements out of the odd components but not the even ones we may now violate (4.3d) and (4.3e).

However, the need to adjust X_e^{2n} after the construction is really only a consequence of our decision to cast the construction as a pinball argument for ease of presentation rather than a Π_2^0 tree construction. By performing this construction in the same fashion as that in 2.1, our ad hoc modification of

X_e^{2n} becomes unnecessary as negative requirements γ can simply delay adding balls to the components of X_e^{2n} until every higher priority requirement \mathcal{R}_e that γ guesses will act infinitely often believes it will not need to add that ball to X_e^{2n} . Whenever a requirement \mathcal{R}_e that γ believes only needs to act finitely many times act γ can simply reset its construction of X_e^{2n} and begin from scratch. Understood in terms of the tree construction the reservation of balls by \mathcal{R}_e acting at γ simply becomes the constraint that any nodes above or to the right of γ cannot pick these elements as new balls. \square

We can prove similarly a version of Lemma 4.2 for the odd case.

Lemma 4.6. *For every $n \geq 1$, there is a properly $2n + 1$ -tardy set A , a c.e. set Z disjoint from A , so that $\hat{A} = A \sqcup Z$ satisfies the conditions of Lemma 4.2. So in particular, \hat{A} is $2n$ -tardy and satisfies all the demands on the enumeration order.*

Proof. One could take a c.e. set \hat{A} built to satisfy Lemma 4.2 and split up the elements we enumerate into \hat{A} into exactly one of the bins A or Z . However, since we must also ensure that A is *properly* $2n + 1$ -tardy it is preferable to dynamically build \hat{A} as above and only once we have made an irrevocable commitment to place x into \hat{A} do we decide whether to put x into A or Z . If we put x into \hat{A} as the result of \mathcal{M}_e , we place x into A . Then, \mathcal{M}_e will be satisfied in the same manner as before. We now focus on those balls used by some requirement \mathcal{R}_e . Without loss of generality, we may assume these balls are not stolen by some higher priority $\mathcal{M}_{e'}$ since we can always react to that event by handing out a new ball to \mathcal{M}_e . As we argued above, eventually \mathcal{M}_e will receive a ball that does not get stolen.

We also place into A any balls that were enumerated into \hat{A} by some \mathcal{R}_e before entering the $2n$ -th component $X_{e_{2n}}^{2n}$ of any node. These balls entered \hat{A} to obtain an immediate victory by showing that either $A \neq B$ or that x does not leave $Y_{e,s}^{2n}$ soon enough before entering B . By placing these balls into A , we ensure that either $Y_{e,s}^{2n}$ does not witness that B is $2n$ -tardy or $A \neq B$. This leaves only the case where x enters $Y_{e_{2n-1},s}^{2n}$ and our construction of \hat{A} responds by starting x down towards the \hat{A} by placing it in the sets $X_{e_{2n}}^{2n}$. As far as the construction of \hat{A} is concerned, once x has entered $X_{e_{2n}}^{2n}$, it must continue on into \hat{A} (modulo possible finite injury). However, when our x reaches the root node we are not forced to place it into A . Instead, we

check if x is still in $Y_{e,s}^{2n}$. If so, we place x into A for the immediate victory. Alternatively if $x \notin Y_{e,s}^{2n}$, i.e., $x \in Y_{e_{2n},s}^{2n}$, then either Y_e^{2n} fails to witness that B is $2n$ -tardy or x enters B so we place x into Z rendering $A = \hat{A} - Z$ properly $2n + 1$ -tardy.

Since the only elements entering \hat{A} but not A pass through all the intermediate components $X_{e_k}^{2n}$ in order at the negative requirements below \mathcal{R}_e , the ordering properties trivially hold at these nodes. At the remaining nodes, x may have become stuck in $X_{j_2}^{2n}$ or some other component. However, this concern is easily addressed by taking any balls we place into Z and slowly running them through the components of X_j^{2n} in order. Then, we can use a slower enumeration of Z to add a final component to X_j^{2n} , making a $2n+1$ -c.e. set that satisfies the hypotheses of the theorem. \square

We now prove Theorem 4.1, i.e., we show that the above sets A in Lemmas 4.2 and 4.6 satisfy $Q_m(A)$. Since the definition of $Q_{2n+1}(A)$ is simply $(\exists Z \subseteq \bar{A})Q_{2n}(A \cup Z)$ and $\hat{A} = A \sqcup Z$ satisfies the conditions of Lemma 4.2, we simply need to show that $Q_{2n}(A)$ holds for A as in Lemma 4.2.

4.2. Verifying Satisfaction

To show that A satisfies $Q_{2n}(A)$ via the C built above we fix some arbitrary $\vec{B} = (B_1, \dots, B_n)$ and will construct $\vec{D} = (D_1, \dots, D_n)$ in response. Furthermore for every $S_j \sqsubset C$ we must also describe the sets $\vec{T}^j = (T_1^j, \dots, T_n^j)$ we will play in response. To gain better control over our construction we fix an effective enumeration $\{(S_j, \hat{S}_j) \mid j \in \omega\}$ containing all disjoint pairs of c.e. subsets of C requiring, by way of the Slowdown Lemma [9] that the indices we list satisfy $S_j \cup \hat{S}_j = C \searrow (S_j \cup \hat{S}_j)$ in the canonical stagewise enumeration of c.e. sets.

We use the tardiness of A to force elements in $(A \cup T_{i+1}) \cap (S_j \cap T_i)$ (where $T_{n+1} = \emptyset$) into B_i whenever $(Q_{2n}(A) : \text{if})$ is satisfied. Since $(Q_{2n}(A) : \text{if})$ forces B_i to copy D_i on $S_j - A$, this occurs automatically for $x \in T_{i+1} - A$. To deal with $x \in A$ we note that if $(Q_{2n}(A) : \text{if})$ is satisfied we can computably measure how long it takes elements in D_i and S_j to appear in either A or B_i . In this case, let $p_j(s)$ be the amount of time it takes for elements entering C before stage s that will eventually enter $S_j \cap D_i$ to enter B_i or A . By tardiness, there exists $X_e^{2n} = \bar{A}$ working faster than this delay p_j . The trick is to then use X_e^{2n} to build D_i as a version of $X_{e_{2i}}$ and T_i^j as a version of $X_{e_{2i-1}}$ so that any x in $A \cap S_j \cap T_i$ has to first enter either B_i or A before entering A and therefore must enter B_i .

To show $Q_{2n}(A)$ holds, we use the $2n$ -c.e. sets X_j^{2n} demonstrating that A is $2n$ -tardy with respect to p_j to construct \vec{D} and \vec{T} . The key point is that, for those elements we care about, D_i will behave like a modified version of some $X_{e_{2i}}^{2n}$ and T_i will behave like a modified $X_{e_{2i-1}}^{2n}$. Thus, \vec{T} can be thought of as elements that may stay out of A and \vec{D} as elements that may enter A . Now, pretending we get to play both \vec{T} and \vec{D} in response to the n -c.e. set X_e^{2n} witnessing tardiness with respect to p_j , we would proceed as follows.

To overcome our inability to build \vec{D} in response to the choice of $S_j \cup \hat{S}_j$ we aim to somehow split up the construction of D_i so that on $T_1^j \cap S_j$ the construction responded to S_j . Unfortunately these sets are not disjoint but the approach remains valid as they make compatible demands. Unfortunately this strategy only works when \vec{D}, \vec{B}, S_j really satisfy $(Q_{2n}(A) : \text{if})$ and even then we must locate the correct $2n$ -c.e. set X_e^{2n} with respect to p_j . We manage this complexity using a Π_2^0 guessing procedure at the true path $f(k)$ described below. Recall that A and C are fixed from Lemma 4.2, and \vec{B} is fixed and arbitrary. We will formally define p_α later.

$$\begin{aligned} f(0) &= j_0 \text{ where } C = W_{j_0} \\ f(2k+1) &= (\mu e)(X_e^{2n} \text{ satisfies Lemma 4.2 with respect to } p_{f(2k+1)}) \\ f(2k+2) &= (\mu j > f(2k)) \left[(S_j \sqcup \hat{S}_j) = C \wedge (Q_{2n}(A) : \text{if}) \text{ holds for } \vec{D}. \right] \end{aligned}$$

We adopt the convention that α, β, γ only range over strings of even length, X^α refers to $X_{e_n}^{2n}$, S_α to S_{j_n} and denote the sets \vec{T} built in response to S_α by \vec{T}^α for $\alpha = (j_0, e_0, j_1, e_1, \dots, j_n, e_n)$. We adopt a standard Π_2^0 approximation argument so that $\alpha \subset f$ iff α is the $<_L$ string of that length satisfying $(\exists^\infty s) [f_s \supseteq \alpha]$. For every x , we keep track of $\Gamma(x, s)$, the leftmost substring of f_t of length x for $t \in [x, s]$ observing that if $\alpha \subseteq f$ then for all but finitely many x we have $\Gamma(x, \infty) \supseteq \alpha$. Before we specify the construction of the function p_α , we detail the construction of \vec{D} and \vec{T}^α .

$$\begin{aligned} x \in T_{1,s+1}^\alpha &\leftrightarrow x \in T_{1,s}^\alpha \\ &\vee [s+1 \geq x \wedge \Gamma(x, s) \supseteq \alpha \wedge x \in X_{1,s}^\alpha - C_s] \end{aligned}$$

$$\begin{aligned}
(4.7a) \quad & s_x = (\mu t)(x \in C_t) \\
& \alpha_x = (\mu \beta \subseteq \Gamma(x, s_x))(\exists t) \\
(4.7b) \quad & \left[x \in T_{1,s_x}^\beta \cap S_{\beta,t} \wedge (\forall \gamma \subset \beta)[x \notin T_{1,s_x}^\gamma \vee x \in \hat{S}_{\gamma,t}] \right] \\
(4.7c) \quad & x \in T_{i+1}^\alpha \leftrightarrow x \in X_{2i+1,s}^{\alpha_x} \cap T_{1,s}^\alpha \\
(4.7d) \quad & x \in D_i \leftrightarrow x \in X_{2i}^{\alpha_x}
\end{aligned}$$

Note that if x is not in T_{1,s_x}^α , it is not in T_1^α . Additionally, given $x \in C_s$ for α on the true path, it is a computable question whether $\alpha_x \subseteq \alpha$ since $S_j \cup \hat{S}_j$ is actually a split of C for $\beta \subseteq \alpha$. We now define $p_\alpha(s)$.

For every x, s define $p_\alpha(i, x, s) = s$ if $x \notin C_s \cup X_{2i,s+1}^{\alpha_x} \vee \alpha_x \not\subseteq \alpha$ otherwise set

$$p_\alpha(i, x, s) = (\mu s' \geq s)(\alpha^- \subseteq f_s \wedge x \in B_{k,s'} \cup A_{s'})$$

Then define

$$p_\alpha(s) = 1 + \max_{\substack{x \leq s \\ i \leq n}} p_\alpha(i, x, s)$$

Lemma 4.8. *If α^- is on the true path then $p_\alpha(s) = p_{\alpha^-}(s)$ is a total function.*

Proof. It suffices to show that $p_\alpha(i, x, s)$ is defined for all $x \leq s$ and $i \leq n$. If $x \notin C_s \cup X_{2i,s+1}^{\alpha_x} \vee \alpha_x \not\subseteq \alpha$, then this is clear. By definition of D_i and α_x , if $x \in X_{2i}^{\alpha_x}$ then $x \in D_i \cap S_{\alpha_x}$. Thus, if $p_\alpha(i, x, s) \neq s$ then as α^- is on the true path ($Q_{2n}(A) : \text{if}$) holds so $x \in A$ or $x \in B_i$. Hence, $p_\alpha(i, x, s)$ is defined by the second clause in its definition. \square

Note the importance of defining p_α using only α^- is to avoid any circularity in selecting the $2n$ -c.e. set witnessing tardiness for p_α at node α .

Lemma 4.9. *If α is on the true path, then for all but finitely many x ,*

$$x \in T_1^\alpha \cap S_\alpha \rightarrow \alpha_x \subseteq \alpha.$$

Proof. For any $x \in T_1^\alpha$, there exists an s such that $\Gamma(x, s) \supseteq \alpha$ with $x \leq s \leq s_x$. Let s' be a stage such that for all $t \geq s'$, f_s is not left of α . Then, for each $x > s'$, we have $s_x \geq s \geq x > s'$ since $x \notin C_{s'}$ by properties of the enumeration of c.e. sets. Hence, for $x > s'$, we have that $\Gamma(x, s_x)$ (and then α_x) is

not to the left of α . Fix $x > s'$ such that $x \in T_1^\alpha \cap S_\alpha$. Since $\Gamma(x, s) \supseteq \alpha$ for $s_x \geq s \geq x > s'$ and α_x is not to the left of α , we have that α_x must either extend α or be a substring of α . Since $x \in T_1^\alpha \cap S_\alpha$, we have that $\alpha_x \not\supseteq \alpha$ by the second conjunct in the definition of α_x . Thus, $\alpha_x \subseteq \alpha$ for all $x > s'$ such that $x \in T_1^\alpha \cap S_\alpha$ as desired. \square

Lemma 4.10. *If α is on the true path, then $A \cap S_\alpha \cap T_i^\alpha \subseteq^* B_i \cap S_\alpha \cap T_i^\alpha$*

Proof. By Lemma 4.9, for all but finitely many $x \in T_1^\alpha \cap S_\alpha$, we have that $\alpha_x \subseteq \alpha$. Take such an x in $A \cap S_\alpha \cap T_i^\alpha$ (Note that $A \cap S_\alpha \cap T_i^\alpha \subseteq T_1^\alpha \cap S_\alpha$.) By definition, $x \in S_{\alpha_x} \cap T_1^{\alpha_x}$. Since $x \in T_i^\alpha$, it follows that $x \in X_{2i-1}^{\alpha_x}$, and since $x \in A$, we also have $x \in X_{2i}^{\alpha_x}$. By (4.3e), if $s+1$ is the least stage such that $x \in X_{2i, s+1}^{\alpha_x}$ then $x \in X_{2i-1, s}^{\alpha_x}$. Furthermore, by (4.3c) we must have $x \in C_s$, so $p_{\alpha_x}(i, x, s)$ is defined by way of the second clause.

By lemma 4.8, we know that $t = p_{\alpha_x}(i, x, s)$ is well defined hence either $x \in A_t$ or $x \in B_t$. However, $p_{\alpha_x}(s) \geq p_{\alpha_x}(i, x, s)$ and $x \in X_{2i-1, s}^{\alpha_x}$ so $x \notin A_t$. Hence $x \in B$. \square

Lemma 4.11. *If α is on the true path then*

$$\left[\begin{array}{l} (A \cup T_2^\alpha) \cap (S_\alpha \cap T_1^\alpha) \subseteq^* B_1 \cap (S_\alpha \cap T_1^\alpha) \\ (A \cup T_3^\alpha) \cap (S_\alpha \cap T_2^\alpha) \subseteq^* B_2 \cap (S_\alpha \cap T_2^\alpha) \\ \dots \\ (A \cup T_n^\alpha) \cap (S_\alpha \cap T_{n-1}^\alpha) \subseteq^* B_{n-1} \cap (S_\alpha \cap T_{n-1}^\alpha) \\ A \cap (S_\alpha \cap T_n^\alpha) \subseteq^* B_n \cap (S_\alpha \cap T_n^\alpha) \end{array} \right]$$

Proof. By Lemma 4.10, the last clause is established. Hence, let x be in $(A \cup T_{i+1}^\alpha) \cap (S_\alpha \cap T_i^\alpha)$ for $i < n-1$ and such that $\alpha_x \subseteq \alpha$ (This is true for all but finitely many of these x by Lemma 4.9.) Again by lemma 4.10 the result is established except for $x \in T_{i+1}^\alpha - A$ so assume this is the case. Since $x \in T_{i+1}^\alpha$ it follows that $x \in X_{2i+1}^{\alpha_x}$ and by (4.3d) we have $x \in X_{2i}^{\alpha_x}$ so $x \in D_i$. But as α on the true path $B_i \cap (S_\alpha - A) = D_i \cap (S_\alpha - A)$ and $x \in D_i \cap (S_\alpha - A)$. Hence $x \in B_i$ completing the proof. \square

Lemma 4.12. *If α is on the true path then*

$$\left[\begin{array}{l} (A \cup T_2^\alpha) \cap (S_\alpha \cap T_1^\alpha) \supseteq^* B_1 \cap (S_\alpha \cap T_1^\alpha) \\ (A \cup T_3^\alpha) \cap (S_\alpha \cap T_2^\alpha) \supseteq^* B_2 \cap (S_\alpha \cap T_2^\alpha) \\ \dots \\ (A \cup T_n^\alpha) \cap (S_\alpha \cap T_{n-1}^\alpha) \supseteq^* B_{n-1} \cap (S_\alpha \cap T_{n-1}^\alpha) \\ A \cap (S_\alpha \cap T_n^\alpha) \supseteq^* B_n \cap (S_\alpha \cap T_n^\alpha) \end{array} \right]$$

Proof. Assume $x \in B_i \cap (S_\alpha \cap T_i^\alpha)$, and suppose that $\alpha_x \subseteq \alpha$ (This holds for all but finitely many $x \in B_i \cap (S_\alpha \cap T_i^\alpha)$ by Lemma 4.9). If $x \in A$, we are done, so suppose not. Since $(Q_{2n}(A) : \text{if})$ is satisfied and $x \in B_i \cap (S - A)$, we must have $x \in D_i$. Thus, by (4.7d) we have $x \in X_{2i}^{\alpha_x}$. If $i = n$, we have that $x \in X_{2n}^{\alpha_x} \subset A$ since $\alpha_x \subseteq \alpha$ is on the true path. This contradicts our original assumption. For $i < n$, since $x \notin A$, we also have that $x \in X_{2i+1}^{\alpha_x}$ by nesting. Since $x \in T_i^\alpha \subset T_1^\alpha$, (4.7c) entails that $x \in T_{i+1}^\alpha$. Hence, $x \in (A \cup T_{i+1}^\alpha) \cap (S_\alpha \cap T_i^\alpha)$, completing the lemma. \square

We now finish demonstrating that A satisfies $Q_{2n}(A)$. Given any \vec{B} , we respond by building \vec{D} as above. Given $S_j \sqsubset C$, we first check whether \vec{B}, \vec{D}, S_j satisfy $(Q_{2n}(A) : \text{if})$, and if not, we are done. If so, there is some finite sequence $\alpha = (j_0, e_0, j_1, e_1, \dots, j_n, e_n)$ along the true path with $j_n = j$, and Lemmas 4.11 and 4.12 above guarantee that $S_j, \vec{T}^\alpha, \vec{B}$ satisfy $(Q_{2n}(A) : \text{then})$ up to \supseteq^* . By construction and Lemma 4.9, $D_{i+1} \subseteq^* D_i$ and $T_{i+1} \subseteq^* T_i$. To ensure that $(Q_{2n}(A) : \text{then})$, $D_{i+1} \subseteq D_i$, and $T_{i+1} \subseteq T_i$ hold exactly, remove the elements in $S_\alpha \cap T_1^\alpha$ that violate the equalities in $(Q_{2n}(A) : \text{then})$ or the subset properties from each T_i^α . Only $T_1 \supseteq \vec{C}$ remains potentially unsatisfied. Note that all elements just removed from T_i^α are elements of C , so these finite modifications do not affect $T_1^\alpha \supseteq \vec{C}$.

If $\alpha \supseteq f$, we have $\Gamma(x, \infty) \supseteq \alpha$ for almost every x and $X_1^\alpha \supseteq \vec{C}$. Hence, $T_1^\alpha \supseteq^* \vec{C}$. Thus, a finite modification of T_1^α suffices to ensure $T_1^\alpha \supseteq \vec{C}$. Since $(Q_{2n}(A) : \text{if})$ only makes demands on $S_\alpha \subseteq C$, these modifications do not affect the satisfaction of $(Q_{2n}(A) : \text{if})$. This completes the proof of satisfaction.

5. A low₂ and simple very tardy

Previously Harrington and Soare established the following theorem in [5].

Theorem 5.1 (Harrington and Soare). *If A is low (or even semi-low) and simple then A is almost prompt.*

Since a very tardy set is simply one that is not almost prompt, this theorem shows that no very tardy can be both low and simple. Harrington and Soare’s proof demonstrates that if A is very tardy and semi-low there is a computable function that grows fast enough so that the corresponding n -c.e. complement of A is forced to leave an infinite c.e. set in \bar{A} . We show that Theorem 5.1 cannot be improved by constructing an example of a low₂ simple very tardy set. This example provides a negative answer to Question 1.6 of Harrington and Soare. We first offer a sketch of the tension in Theorem 5.1 so as to motivate the construction of our example.

Building a low set requires that we eventually preserve computations of the form $\Phi_e(A_s; e)$, while simplicity requires that if $W_{i,s}$ continues to grow, we eventually enumerate one of its members into A . Normally, we build a low simple set by only allowing a finite number of computations $\Phi_e(A_s; e)$ to restrain elements we see enter W_i out of A . However, building a very tardy set requires that we announce our intention to enumerate some element y into A long in advance. During the intervening time, a computation $\Phi_e(A_s; x)$ might converge and impose a restraint that y is obligated to respect. Hence, y must abandon its previously announced intention to enter A . If A was meant to be 2-tardy, this alone would cause a failure since 2-tardy sets cannot revoke their announced intentions to place elements into A . It might seem, on the other hand, that if we only aim to build a very tardy we could simply choose to leave y out of A and wait for another chance to place an element from W_i into A . However, by the time we observe that some y_n enters W_i , some later W_{i_n} may have already attempted to enumerate y_n into A and abandoned that attempt in response to a restraint from some computation $\Phi_{e_n}(A_s; e_n)$. Indeed, each y_n entering W_i may have already exhausted its guesses about entering A so that W_i no longer has the opportunity to place y into A .

It is clear from the above discussion that the need to restrain elements from entering A creates the potential to ‘use up’ the elements of some infinite c.e. set before we have the chance to place one of its members into A . Since lowness requirements in general require imposing some kind of restraint, it is interesting to see that Harrington and Soare’s result fails for a weaker notion of lowness.

Theorem 5.2. *There is a simple 2-tardy set A with $A'' = \mathcal{Q}''$*

For the sake of readability, we denote the e th partial computable function Φ_e as p_e when regarded as a potential total computable function p as in Definition 1.7.

Our construction will satisfy a version of the following two requirements.

$$\begin{array}{ll}
\mathcal{N}_e & p_e \text{ is partial or there is a 2-c.e. set } X_e^2 = \bar{A} \text{ s.t. } X_{e,s}^2 \cap A_{p(s)} = \emptyset \\
\mathcal{P}_e & |W_e| = \infty \implies A \cap W_e \neq \emptyset \\
\mathcal{R}_e & \text{If } \limsup_s |W_{e,s}^A| = \infty \implies |W_e^A| = \infty
\end{array}$$

Requirement \mathcal{P}_e guarantees that A is simple, and \mathcal{N}_e ensures that A is 2-tardy. Notice that

$$\limsup_s |W_{e,s}^A| = \infty \iff |\{ \langle s, n \rangle \mid s = \mu t \ (|W_{e,t}^A| \geq n) \}| = \infty$$

This fact and the fact that $\{e \mid |W_e^A| = \infty\} \equiv_{\mathbf{T}} A''$ guarantee that $A'' = \mathcal{Q}''$. Our actual construction will take place on a tree. Taking advantage of this tree structure, however, will require a minor modification of \mathcal{R}_e .

5.1. Tree Argument

For every element $\alpha \in 2^{<\omega}$, we tentatively associate a module \mathcal{Q}_α that tries to implement an associated requirement according to the following rule.

$$\mathcal{Q}_\alpha \text{ implements } \begin{cases} \mathcal{N}_e & \text{if } |\alpha| = 3e \\ \mathcal{P}_e & \text{if } |\alpha| = 3e + 1 \\ \mathcal{R}_e & \text{if } |\alpha| = 3e + 2 \end{cases}$$

In the interest of clarity, we simply write $\mathcal{N}_\alpha, \mathcal{P}_\alpha$ or \mathcal{R}_α to refer to the module \mathcal{Q}_α in the case that it is associated with $\mathcal{N}_e, \mathcal{P}_e$ or \mathcal{R}_e , respectively. Conversely, we use \mathcal{Q}_n to denote the requirement implemented by the modules \mathcal{Q}_α with $|\alpha| = n$. As a notational convenience, we write p_α and W_α in place of p_e and W_e when discussing $\mathcal{N}_\alpha, \mathcal{P}_\alpha$ and \mathcal{R}_α . We say \mathcal{Q}_α is satisfied to indicate that \mathcal{Q}_α satisfies $\mathcal{Q}_{|\alpha|}$.

We define the true path function, $f : \omega \mapsto \{0, 1\}$ with the property that if $\alpha \subseteq f$ then the module \mathcal{Q}_α satisfies $\mathcal{Q}_{|\alpha|}$. In an abuse of notation, we also view f as a function from $f : 2^{<\omega} \mapsto \{0, 1\}$ where $f(\alpha)$ indicates the manner in which $\mathcal{Q}_{|\alpha|}$ is satisfied so that $f(|\alpha| + 1) = f(|\alpha|) \hat{\ } f(\alpha)$.

At any given stage s , we will have an approximation $f_s \in 2^{<\omega}$ to f where we similarly abuse notation and write $f_s(\alpha)$ to denote $f_s(|\alpha|)$ provided $f_s \supseteq \alpha$. At a given stage s , only those modules lying along the approximation f_s receive attention (are visited) so in what follows we always assume that $\alpha \subseteq f_s$ when describing \mathcal{Q}_α . If α is not visited at stage s , then we stipulate that $f_s(\alpha)$ is undefined. Thus, if α is the \subset -maximal node visited at s , then $f_s = \alpha \hat{\ } f_s(\alpha)$. The construction will ensure that $f = \liminf_s f_s$ and the action of each module will be described uniformly in α so that $f_s \in 2^{<\omega}$ is a computable function of s .

5.1.1. Motion On The Tree

Various numbers, called balls, will be located on our nodes. The function $\alpha(x, s)$ equals the node occupied by x at stage s or \uparrow if x is not on the tree. One should think of the tree as growing upward with elements trickling down the tree towards A during the course of the construction. Any element reaching λ is immediately placed into A , so $\alpha(x, s) = \lambda$ if and only if $x \in A_s$ after which point $\alpha(x, s)$ it can no longer change value.

At a given stage s , the construction will begin by executing \mathcal{Q}_λ . Suppose the module \mathcal{Q}_β receives attention and sets $f_s(\beta) = i$. If, for all $|\beta| < s$, requirement \mathcal{Q}_β did not enumerate a ball onto the tree and no ball occupies any $\beta' \subseteq \beta \hat{\ } i$, then $\mathcal{Q}_{\beta \hat{\ } i}$ receives attention. This condition ensures that whenever \mathcal{Q}_α is executed, every ball below α on the tree is already in A . We say that the node β is visited at stage s if the module \mathcal{Q}_β is executed at stage s . If β is visited at stage s and $f_s(\beta) = 0$, we say that β is expansionary at stage s . We write $\alpha <_L \beta$ to indicate that for some l , node $\alpha \upharpoonright_l$ occurs lexicographically before $\beta \upharpoonright_l$ and if $f_s <_L \beta$, we say that β (equivalently \mathcal{Q}_β) is reset at stage s .

If $\alpha(x, s)$ is reset at stage s , then x is removed from the machine and $\alpha(x, s)$ is undefined. Intuitively, this corresponds to abandoning our plans regarding x because our approximation to the true path moved to the left of $\alpha(x, s)$. Since the \mathcal{N}_β modules are the only modules that wish to delay the entry of elements into A , we only allow balls to occupy nodes of the form $\beta \hat{\ } 0$ where \mathcal{Q}_β implements \mathcal{N}_β . That is, balls flow down towards A until they are restrained by some delay function p_β . We ensure by definition that whenever f_s extends $\alpha(x, s-1) = \beta \hat{\ } 0$, the restraint imposed by \mathcal{N}_β is released and the ball is moved to the \subset -maximal node $\nu \hat{\ } 0 \subset \beta$ with \mathcal{Q}_ν implementing \mathcal{N}_ν or into A if no such ν exists.

We write:

$$s_\alpha = \max_{t \leq s} \{t \mid \alpha \text{ is visited at stage } t\}$$

$$\mathfrak{r}(\alpha, s) = \max_{t \leq s} \{t \mid \mathcal{Q}_\alpha \text{ is reset at stage } t\}$$

Any \mathcal{P}_α for $\alpha \supseteq \beta \hat{\ } 0$ can place elements on the tree at an allowed node $\beta \hat{\ } 0$. However, at stage s , \mathcal{P}_α is only allowed to place a ball x onto the tree provided $\mathfrak{r}(\alpha, s) < x \leq s$. Note that this implies that there is no $t < s$ with $\alpha(x, t) <_L \alpha$. This ensures that, when reset, a node starts with a fresh set of balls. In other words, nodes cannot recycle balls that have been to their left, or equivalently, a node can recycle a ball only when that node has never been in a position to notice that the ball was used previously. We adopt the convention that the use of $W_{\alpha, s}$ is no more than s . Hence, the action of the machine ensures that if α is visited at stage s then no α' with $\alpha <_L \alpha'$ can disrupt this computation because α' was reset and any new balls that might be placed in for α' will be greater than s .

The requirements \mathcal{R}_α for $\beta \subseteq \alpha$ will work together to define a restraint function $r(\beta, s)$, and \mathcal{P}_β will only be allowed to add a ball x to the machine only if $x > r(\beta, s)$. More precisely, each \mathcal{R}_α will have its own restraint function $r_\alpha(\cdot)$, and we define

$$r(\beta, s) = \max_{\alpha \subset \beta} r_\alpha(\beta, s)$$

If neither the action of the tree nor the restraint function above bars \mathcal{P}_β from placing x on the tree at stage s , then we say that x is available to β at stage s .

5.1.2. The \mathcal{N}_α module

If \mathcal{Q}_α implements \mathcal{N}_α then we define

$$f(\alpha) = \begin{cases} 0 & \text{if } p_\alpha \text{ total} \\ 1 & \text{if } p_\alpha \text{ partial} \end{cases}$$

$$l^\alpha(s) = \mu z (p_{\alpha, s}(z) \uparrow)$$

$$f_s(\alpha) = \begin{cases} 0 & \text{if } l^\alpha(s) > \max[t < s \mid f_t \supseteq \alpha \hat{\ } 0] \\ 1 & \text{otherwise} \end{cases}$$

Recall that if $f_s(\alpha) = 0$, we call s an expansionary stage. As long as it is not reset, \mathcal{N}_α builds the set X_α^2 as follows. At stages where $f_s(\alpha) = 0$, every $x < s$ with $\alpha(x, s) \not\subseteq \alpha$ and $x \notin A_s$ is enumerated into $X_{\alpha_1}^2$. Also, whenever an element x is placed at $\alpha \hat{\ } 0$, it is enumerated into $X_{\alpha_2}^2$. The net effect of the definition of f_s is that if t is the last stage where some ball may have been placed at $\alpha \hat{\ } 0$ then $f_s(\alpha) \neq 0$ unless $p_\alpha(t) \downarrow_s$. Thus, the ball is not released until the delay demanded by \mathcal{N}_α after entering $X_{\alpha_2}^2$ has expired. If \mathcal{N}_α is reset, we reset the sets $X_{\alpha_1}^2$ and $X_{\alpha_2}^2$. Specifically, at the next stage α is visited, we initialize $X_{\alpha_1}^2 = \{x < s \mid \alpha(x, s) \not\subseteq \alpha\}$ and $X_{\alpha_2}^2 = \emptyset$ before continuing as usual.

5.1.3. The \mathcal{P}_α module

If \mathcal{Q}_α implements \mathcal{P}_α then we define

$$f(\alpha) = \begin{cases} 0 & \text{if } W_\alpha \cap A \neq \emptyset \\ 1 & \text{if } W_\alpha \subseteq \bar{A} \text{ and finite} \end{cases}$$

$$f_s(\alpha) = \begin{cases} 0 & \text{if } A_s \cap W_{\alpha,s} \neq \emptyset \\ 1 & \text{otherwise} \end{cases}$$

The action of \mathcal{P}_α tries to place some element from W_α into A . When there is an x available for α already in $W_{\alpha,s}$, requirement \mathcal{P}_α places x on the largest node $\beta \hat{\ } 0 \subset \alpha$ with \mathcal{Q}_β implementing a negative requirement.

5.1.4. The \mathcal{R}_α module

If \mathcal{Q}_α implements \mathcal{R}_α , then we define

$$f(\alpha) = \begin{cases} 0 & \text{if } \limsup_s |W_{e,s_\alpha}^A| = \infty \\ 1 & \text{if } \limsup_s |W_{e,s_\alpha}^A| < \infty \end{cases}$$

$$l^\alpha(s) = |W_{e,t}^{A_t}| \text{ where } t = s_\alpha$$

For $n \leq l^\alpha(s)$,

$$u_s^\alpha(n) = \mu t \leq s (\exists x_1 \neq x_2 \neq \dots \neq x_n) (\forall j \leq n) (\forall t' \in [t, s]) [x_j \in W_{\alpha,t'}^{A_{t'}}]$$

$$f_{s+1}(\alpha) = \begin{cases} 0 & \text{if } l^\alpha(s+1) > l^\alpha(s) \\ 1 & \text{otherwise} \end{cases}$$

The role of \mathcal{R}_α is to define a restraint function that preserves computations of the form $|W_{\alpha,s}^A| = n$. To this end, if α is visited at s then set

$$r_\alpha(\beta, s) = \begin{cases} 0 & \text{if } \beta \not\preceq \alpha \hat{=} 0, \text{ otherwise} \\ s & \text{if } |\beta| - |\alpha| > l^\alpha(s) \\ u_s^\alpha(|\beta| - |\alpha|) & \text{else} \end{cases}$$

Notice that no restraint is placed on nodes $\beta \not\preceq \alpha \hat{=} 0$. Our convention about the use of $|W_{\alpha,s}^A|$ and the action of the tree guarantees that no \mathcal{P}_β for $\beta \not\preceq \alpha \hat{=} 0$ is able to remove elements from $W_{\alpha,s}$. On the other hand, the restraint function ensures that there are only finitely many nodes that might disrupt the computations responsible for enumerating the n longest residing elements of $W_{\alpha,s}$ for $n \leq l^\alpha(s)$. We will demonstrate below that this ensures a variant of \mathcal{R}_α is satisfied.

5.2. Verification

5.2.1. Tree Requirements

The construction above does not quite satisfy the requirements given above. Instead, for α along the true path, we show the following modified versions of the requirements are satisfied.

$$\begin{aligned} \mathcal{N}_\alpha & \quad p_\alpha \text{ is partial or there is a 2-c.e. set } X_\alpha^2 =^* \bar{A} \text{ s.t. } X_{\alpha,s}^2 \cap A_{p(s)} = \emptyset. \\ \mathcal{P}_\alpha & \quad |W_\alpha| = \infty \implies A \cap W_\alpha \neq \emptyset. \\ \mathcal{R}_\alpha & \quad \limsup_{s \rightarrow \infty} l^\alpha(s) = \infty \implies |W_\alpha^A| = \infty. \end{aligned}$$

The requirements \mathcal{N}_α and \mathcal{P}_α differ only in notation. The requirement \mathcal{R}_α differs in that $\limsup_s |W_{\alpha,s}^A|$ is only evaluated at stages s such that α is on the true path. This avoids any transitory effects that might enumerate elements into $W_{\alpha,s}^A$ and then remove them again before \mathcal{R}_α has a chance to act. As we will see, however, it is no less effective a means to show that $A'' = \mathcal{Q}''$.

5.2.2. Satisfaction

The essential property satisfied by a Π_2^0 tree construction is:

Lemma 5.3. $f = \liminf_s f_s$

Recall that we write $f_s(\alpha)$ to denote $f_s(|\alpha|)$ provided $f_s \supseteq \alpha$. In our construction, Lemma 5.3 can be verified by straightforward inspection of the modules where we understand $f(n)$ only to be defined if $f_s(n)$ is defined for infinitely many s . We will later show that f is a total function but first we observe three important properties of our construction.

Lemma 5.4.

1. If \mathcal{Q}_α is visited at stage s , there are no balls at any $\beta \subset \alpha$ not in A .
2. If $f_s \supseteq \alpha(x, s-1)$ then $\alpha(x, s) \subsetneq \alpha(x, s-1)$, provided $\alpha(x, s-1)$ and $\alpha(x, s)$ are defined.
3. If α is on the true path, then there is a stage s after which \mathcal{Q}_α is never reset.

The first two follow directly from the action of the tree, and the third property follows immediately from Lemma 5.3. We now show that the true path is total. We first note that no individual ball ever stalls on the true path.

Lemma 5.5. *If $\alpha(x, s) = \alpha$ and $f_t \supseteq \alpha$ where $t > s$, then either \mathcal{Q}_α was reset between stage t and s or $x \in A_t$.*

Proof. The only way x can leave α is if α is reset or x moves to a predecessor of α . By induction, x must either reach A or α is reset when its predecessor is reset. \square

We need to know that no positive requirement emits so many balls that f_s cannot extend a particular node.

Lemma 5.6. *If $\alpha \subseteq f$ then \mathcal{P}_α places at most finitely many balls on the tree. Furthermore, if $\alpha \subseteq f$ and $\beta \supset \alpha$, then only finitely many balls placed on the tree by \mathcal{P}_β travel down the tree to reach α .*

Proof. If α places infinitely many balls on the tree, then there is a stage s and a ball $x \in W_{\alpha, s}$ so that \mathcal{P}_α emits x at stage s and α is no longer reset after stage s . Furthermore, \mathcal{P}_α must emit another ball later, so there is a stage $t > s$ with $f_t \supseteq \alpha$. Hence, $x \in A_t$ by 5.5. By the action of the \mathcal{P}_α module, however, \mathcal{P}_α ceases emitting once $x \in A_t \cap W_{\alpha, t} \neq \emptyset$. Thus, only finitely many balls are placed on the tree by \mathcal{P}_α . The second half of the claim follows by the same argument applied to the stages at which x reaches α . \square

Lemma 5.7. *The function f is total, and every $\alpha \subset f$ is visited infinitely many times.*

Proof. Suppose not; then $f = \alpha$ for some $\alpha \in 2^{<\omega}$ and $\{s \mid f_s = \alpha\}$ is infinite. Without loss of generality, we may assume that f_s never properly extends α for $s > 0$. By construction, this occurs for large s only if every time $f_s = \alpha$, there is some x with $\alpha(x, s) = \alpha^-$. Pick s_0 large enough so that α is never reset after s_0 , and no \mathcal{P}_β with $\beta \subseteq \alpha$ places any ball on the tree after s_0 . Now, pick some t and s_1 where $t > s_1 > s_0$ such that $f_t = f_{s_1} = \alpha$ and no nodes above α are visited between stages t and s_1 . If x is such that $\alpha(x, t) = \alpha^-$, the ball x cannot have trickled down from above nor can it have been placed at α^- after s_0 . Hence, $\alpha(x, s_1) = \alpha^-$, violating Lemma 5.5, a contradiction. \square

Before we can conclude that \mathcal{P}_α is satisfied, we first must argue that \mathcal{R}_α imposes only finitary restraint on the true path. We need two further lemmas. The first shows that the only way ball x can pass by ball y is if y is placed on the tree first and later x is added with $y <_L x$.

Lemma 5.8. *Suppose x is placed on the tree at stage s and $\alpha(y, s) \not\supseteq \alpha(x, s)$. If x remains on the tree until $s' > s$ and $\alpha(y, s') \supseteq \alpha(x, s')$, then $\alpha(y, s) <_L \alpha(x, s)$.*

Proof. Since the construction guarantees that the nodes above the position of x are not visited, y cannot be added to the tree above x . Since balls move downward on the tree, if $t + 1 > s$ is the least stage at which $\alpha(y, t + 1) \supseteq \alpha(x, t + 1)$, we have either $\alpha(x, t) <_L \alpha(y, t)$ or $\alpha(y, t) <_L \alpha(x, t)$. However, the motion of x at stage t requires that $f_{t+1} \supseteq \alpha(x, t)$. So, if $\alpha(x, t) <_L \alpha(y, t)$, then y is removed from the tree at stage $t+1$, contradicting our assumption that $\alpha(y, t + 1) \supseteq \alpha(x, t + 1)$. Therefore, $\alpha(y, t) <_L \alpha(x, t)$. Provided that the balls remain on the tree between s and t , we have $\alpha(y, t) \subseteq \alpha(y, s)$ and $\alpha(x, s) \supseteq \alpha(x, t)$. If y was added to the tree to the left of the location of x after x was placed on the tree, then x would have been removed from the tree. So, $\alpha(y, s) <_L \alpha(x, s)$. \square

We can now infer that if α is on the true path, then infinitely often the only balls above α are those that will never move below α without being reset. Let

$$B_t = \{x \mid \alpha(x, t) \supseteq \alpha \wedge (\exists t' > t)(\alpha(x, t') \subseteq \alpha)\}.$$

Let \hat{B}_t be the set of elements in B_t that reach α without being reset between stages t and t' in the definition of B_t .

Lemma 5.9. *If $\alpha \subset f$ then for every s there is a $t \geq s$ such that $f_t \supset \alpha$ and $\hat{B}_t = \emptyset$.*

Proof. Without loss of generality, we may assume s is so large that α is never reset after s and that every positive requirement below α no longer acts. Suppose $\hat{B}_s \neq \emptyset$. Let $x \in \hat{B}_s$ be such that $\alpha(x, s)$ is minimal in \hat{B}_s under $<_L$ and maximal among those elements under \supset . By definition of B_s , there is a least stage $t > s$ at which $\alpha(x, t) \subseteq \alpha$. Now, given any $y \in B_t$, we have $\alpha(y, t) \supseteq \alpha \supset \alpha(x, t)$, and by maximality of x under \subset , we know that $\alpha(y, s) \not\supseteq \alpha(x, s)$. So, by Lemma 5.8, we know that $\alpha(y, s) <_L \alpha(x, s)$. Since neither x nor y was reset between stages s and t , the element y is an element in \hat{B}_s to the left of x , a contradiction. Hence, $\hat{B}_t = \emptyset$. The second part of the claim follows by choosing $t' \geq t$ to be the stage at which x enters A and again applying the previous lemma to show nothing from the right could get above x . \square

Recall $r(\beta, s) = \max_{\alpha \subset \beta} r_\alpha(\beta, s)$ is the restraint imposed on \mathcal{P}_β by \mathcal{R}_β .

Lemma 5.10. *If α is on the true path, then \mathcal{R}_α is satisfied, and if $\beta \supset \alpha$ is on the true path, then $\lim_{s \rightarrow \infty} r_\alpha(\beta, s)$ is finite. Hence, $\lim_{s \rightarrow \infty} r(\beta, s)$ is finite as well.*

Proof. If $\limsup_s l^\alpha(s) < \infty$, then \mathcal{R}_α is satisfied, and the second half follows trivially (since $\beta \not\supseteq \alpha \hat{\ } 0$). Otherwise, we work after a stage large enough such that α is not reset anymore and no positive requirements below α place balls on the tree. We claim that for every n there is some stage s_n and elements x_1, x_2, \dots, x_n satisfying

1. $f_{s_n} \supseteq \alpha$,
2. $(\forall i \neq j \leq n)[x_i \neq x_j]$, and
3. $(\forall i \leq n)(\forall t \geq s_n)[x_i \in W_{\alpha, t}^{A_t}]$.

By the definition of $u_s^\alpha(n)$, the last equation entails that $s_n \geq u_t^\alpha(n)$ since element $x_n \in W_{\alpha, t}^{A_t}$ for every $t \geq s_n$.

To verify the claim, suppose n is the least failure of this claim. Pick $s > s_{n-1}$ large enough such that every ball placed on the machine by \mathcal{P}_β with $\beta \supset \alpha$ and $|\beta| - |\alpha| < n$ which will ever enter A has already done so and

at which $l^\alpha(s) < n$. Now pick $s' > s$ as given in Lemma 5.9 and $t \geq s'$ least with $f_t \supseteq \alpha \hat{0}$. Note that $\hat{B}_t = \emptyset$ as well. At stage t , the only balls above α are those added at this very stage and those that will be reset before they get below α .

By the strategy given for \mathcal{R}_α , we know that $l^\alpha(t) > l^\alpha(t-1) \geq n-1$. Let x_n be the element that has occupied $W_{\alpha,t}^{A_t}$ for the longest uninterrupted time and is not equal to any of x_m for $m < n$. Since $f_t <_L \alpha \hat{1}$, no \mathcal{P}_β with $\beta \supseteq \alpha \hat{1}$ can add balls less than t to A and, thus, cannot remove x_n from W_α^A . On the other hand, if $\beta \supseteq \alpha \hat{0}$ and if \mathcal{P}_β is to succeed in placing any elements in A , we must have $|\beta| - |\alpha| \geq n$. This implies that, for all $t' \geq t$, if $x_n \in W_{\alpha,t'}^{A_{t'}}$ and \hat{s} is the first stage at which x_n entered W_α^A and remained in until t' , then $r_\alpha(\beta, t') \geq \hat{s}$. Thus, the restraint guarantees any new balls placed on the tree cannot remove x_n from W_α^A . Moreover, any old balls already on the tree above α are reset before they get to α because $\hat{B}_t = \emptyset$, so they cannot remove x_n from W_α^A . This verifies the claim, and the lemma follows immediately. \square

Lemma 5.11. *A is 2-tardy and simple.*

Proof. The sets built at the \mathcal{N}_α nodes along the true path witness that A is 2-tardy. Suppose p_α is total. Let s' be the last stage at which α is reset. For each stage $s \geq s'$ such that $f_s \supset \alpha$, all elements $x < s$ with $x \notin A_s$ for which $\alpha(x, s) \not\subseteq \alpha$ are enumerated into $X_{\alpha_1}^2$. By construction, any ball in $X_{\alpha_1}^2$ that passes through node α after stage s' enters $X_{\alpha_2}^2$ and is appropriately delayed by p_α before entering A . Consider a ball y on a node $\beta' \supseteq \alpha'$ for $\alpha' \subset \alpha$ and $\beta' >_R \alpha$ at some stage $t > s'$ where $\beta \subset f_t$. Let t' be the least stage greater than t such that $f_{t'} \supset \alpha$. If y is on a node to the right of α at stage t' , the ball y is recycled (and hence is not added to $X_{\alpha_1}^2$ at this stage). Otherwise, y has been reset or has already entered A by stage t' via the node α' without being placed in $X_{\alpha_1}^2$ or $X_{\alpha_2}^2$. By Lemma 5.10, if W_β is an infinite c.e. set, then eventually some element in W_β is greater than the finite value $\lim_{s \rightarrow \infty} r(\beta, s)$ and enters W_β after the last stage at which β is reset. Thus, \mathcal{P}_β will succeed in making $A \cap W_\beta \neq \emptyset$. \square

Lemma 5.12. $A'' \leq_{\mathbf{T}} \mathcal{Q}''$

Proof. Recall that requirement \mathcal{R}_α guarantees that $W_{|\alpha|}^A$ is infinite if and only if $\limsup_s l^\alpha(s) = \infty$. To determine whether $W_{|\alpha|}^A$ is infinite, \mathcal{Q}'' simply computes $f(\alpha)$. Thus, $A'' \leq_{\mathbf{T}} \mathcal{Q}''$. \square

This completes the proof of the theorem.

6. Open Questions

First, we would like to know whether there are any n -tardy sets that are not automorphic to a complete set that do not satisfy Q_n .

Question 6.1. *Does every n -tardy set not automorphic to a complete set satisfy Q_n ?*

We would also like to know whether there are properly very tardy sets that are not automorphic to a complete set.

Question 6.2. *Is there a very tardy set that is not n -tardy for any n and is not automorphic to a complete set?*

The above question could be attacked using definable properties. We have not yet found a property that describes the properly very tardy sets, i.e., those very tardy sets that are not n -tardy for any $n \in \omega$.

Question 6.3. *Find a property Q_∞ so that if $Q_\infty(A)$ holds, then A is very tardy, and find some very tardy set A that is not n -tardy for any n and satisfies $Q_\infty(A)$.*

Finally, we want to know whether Theorem 2.1 can be extended as follows.

Question 6.4. *Is there a properly $n + 1$ -tardy set that is not computed by any n -tardy sets?*

- [1] P.A. Cholak, R. Downey, M. Stob, Automorphisms of the lattice of recursively enumerable sets: promptly simple sets, *Trans. Amer. Math. Soc.* 332 (1992) 555–570.
- [2] R.M. Friedberg, Two recursively enumerable sets of incomparable degrees of unsolvability (solution to Post’s problem, 1944), *Proc. Nat. Acad. Sci. U.S.A.* 43 (1957) 236–238. [2](#)
- [3] L. Harrington, R.I. Soare, Definability, automorphisms, and dynamic properties of computably enumerable sets, *Bull. Symbolic Logic* 2 (1996) 199–213. [2](#), [5](#)

- [4] L.A. Harrington, R. Soare, Post's program and incomplete recursively enumerable sets, *Proceedings of the National Academy of Sciences* 88 (1991) 10242–10246. [2](#), [4](#)
- [5] L.A. Harrington, R.I. Soare, The Δ_0^3 -automorphism method and noninvariant classes of degrees, *J. Amer. Math. Soc.* 9 (1996) 617–666. [2](#), [3](#), [5](#), [30](#)
- [6] L.A. Harrington, R.I. Soare, Codable sets and orbits of computably enumerable sets, *J. Symbolic Logic* 63 (1998) 1–28. [2](#), [4](#), [5](#), [13](#), [14](#)
- [7] A.A. Mučnik, On the unsolvability of the problem of reducibility in the theory of algorithms, *Dokl. Akad. Nauk SSSR (N.S.)* 108 (1956) 194–197. [2](#)
- [8] E.L. Post, Recursively enumerable sets of positive integers and their decision problems, *Bull. Amer. Math. Soc.* 50 (1944) 284–316. [1](#)
- [9] R.I. Soare, *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets, Perspectives in Mathematical Logic*, (Springer-Verlag, Berlin, 1987). [2](#), [15](#), [16](#), [26](#)