November 20, 2006

SLENDER CLASSES.

ROD DOWNEY AND ANTONIO MONTALBÁN

Abstract. A Π^0_1 class P is called *thin* if, given a subclass P' of P there is a clopen C with $\mathcal{P}' = P \cap C$. Cholak, Coles, Downey and Herrmann [7] proved that a Π^0_1 class P is thin if and only if its lattice of subclasses forms a Boolean algebra. Those authors also proved that if this boolean algebra is the free Boolean algebra, then all such think classes are automorphic in the lattice of Π^0_1 classes under inclusion. From this it follows that if the boolean algebra has a finite number n of atoms then the resulting classes are all automorphic. We prove a conjecture of Cholak and Downey [8] by showing that this is the only time the Boolean algebra determines the automorphism type of a thin class.

1. Introduction

A (computably bounded) Π^0_1 class C can be defined as the set of infinite paths through a computable tree $T \subseteq 2^{<\omega}$. The study of Π^0_1 classes has a long and interesting history, and many applications. These applications include those to effective model theory (e.g. Jockusch and Soare [13]), combinatorics (e.g. Remmel [19]), proof theory (through the use of the low basis theorem and the like), and more recently effective randomness (such as in Nies, Stephan and Terwijn [17]). We refer the reader to the surveys Cenzer [2], Cenzer-Remmel [6], Cenzer-Jockusch [4] and Simpson [20].

This paper continues the study of the lattice of Π^0_1 classes along the lines of Cenzer, Downey, Jockusch, and Shore [3]. We are interested in the class of Π^0_1 classes introduced in Downey [9, 10], but first constructed under duality in Martin and Pour-El [16]. These are the *thin* classes, where an infinite class P is called thin if, for all Π^0_1 subclasses $P' \subseteq P$ there is a clopen set C such that $C \cap P = P'$.

Thin classes have attracted considerable interest, and have particularly interesting degree-theoretical properties, as well as significant connections with algorithmic randomness such as Simpson [20], and Binns [1].

Thin classes more or less correspond to hyperhypersimple sets in the lattice of computably enumerable sets. This intuition was made clear in Cholak, Coles, Downey, and Herrmann [7], where it is proven that an infinite class P is thin if and only if the lattice of subclasses forms a Boolean algebra which is always Δ^0_2, and every Δ^0_2 Boolean algebra is isomorphic to the lattice of subclasses of some thin class. This characterization can be viewed as the analog of Lachlan's result [14] that the collection of computably enumerable supersets of a hyperhypersimple set is a Σ^0_3 Boolean algebra, and every such Boolean algebra can be realized.

2000 *Mathematics Subject Classification.* 03D25, 03D28,03D45.

Key words and phrases. thin, slender, Π^0_1-classes, automorphisms, Boolean algebras.

The first author's research was partially supported by The Marsden Found of New Zealand. The second author was partially supported by NSF Grant DMS-0600824 and by the the Marsden Found of New Zealand.

The main result of [7] is that if S and T are *perfect* thin classes, then there is an automorphism of the lattice of Π^0_1 classes under inclusion taking S to T. Here the class is perfect if and only if the lattice of subclasses is isomorphic to the free (also known as atomless) Boolean algebra, or, equivalently the class has no isolated points. (Additionally, it is also proven in [7] that the degrees of such classes are exactly the *array non-computable degrees* of Downey, Jockusch and Stob [11]. Thus these classes and their degrees correspond to a Π^0_1 class analog of Soare's result [21] that maximal sets form an orbit, and Martin's one [15] that the maximal sets all have high degrees.)

After seeing [7], it seemed reasonable to suggest that if the Boolean algebras of subclasses of two thin classes were isomorphic, then the classes would be automorphic. For example, the method of proof of [7] would show that if the Boolean algebra of subclasses of two thin classes T_1 and T_2 have the same finite number of atoms, then T_1 and T_2 are automorphic.

This hope was shown to fail in general by Cholak and Downey [8] who showed it failed for *minimal classes*. Here a thin class M is called minimal if M has a unique non-isolated (rank one) point in it, and hence every Π^0_1 subclass is either finite or cofinite in M. These were first introduced in [3]. Cholak and Downey [8] formulated a new (definable) property, *cohesive* minimality, and proved that there are minimal classes which are cohesively minimal, and there are minimal classes which are not cohesively so. Thus, whilst the classes have the same lattices of subclasses (the Boolean algebra of finite and cofinite subsets), they cannot lie in the same orbit.

After proving this result, Cholak and Downey offered the following conjecture.

Conjecture 1.1 (Cholak and Downey [8])**.** The only Boolean algebras $\mathcal{B}$, which have the property that any two thin classes with $\mathcal{B}$ as their lattices of subclasses are automorphic, are ones with a finite number of atoms.

In this paper we will prove this conjecture.

Theorem 1.2. *Suppose that $\mathcal{B}$ is a Δ^0_2 Boolean algebra with infinitely many atoms. Then there are thin classes T_1 and T_2 both having $\mathcal{B}$ as their lattices of subclasses, and such that T_1 and T_2 are not automorphic.*

The idea of the proof of this result is a generalization of that used by Cholak and Downey. Let $P \subseteq 2^\omega$ be a Π^0_1 class. Then we will denote by $int(P) \subseteq 2^\omega$ the interior of P (i.e. the largest open subset of P), and by $iso(P) \subseteq 2^\omega$ the set of isolated points of P.

Definition 1.3. A Π^0_1 class S is *slender* if for every other Π^0_1 class F, there exists a clopen set C such that

$$int(F) \cap iso(S) = C \cap iso(S)$$

Our main result will follow once we have proven that there are T_1, T_2 as above, one of which is slender and one of which is not. Actually this difference will be elementary because of work of Cenzer and Nies [5]. As with Cholak and Downey [8], the method of proof is a "full approximation" construction for the Π^0_1 classes. However, the fact, for example, that there are Δ^0_2 Boolean algebras with no computable presentations, and for whom the atoms are not Δ^0_2, means that there are many further layers of complexity within the proof. This necessitates a kind of non-uniformity to the strategies within the proof according to whether we are in some dense bit or not, as the reader will see in Section 4. Some of the technical difficulties are solved using algebra. That is, some these difficulties are allayed by the use of a topological form of the Remmel-Vaught Theorem (from [18]) which states that if $\mathcal{B}_1$ and $\mathcal{B}_2$ are Boolean algebras with infinitely many atoms, and $\mathcal{B}_1$ results from $\mathcal{B}_2$ by taking $\mathcal{B}_2$'s atoms $\{b_i : i \in \omega\}$ and splitting each b_i into finitely many atoms, then $\mathcal{B}_1$ and $\mathcal{B}_2$ are isomorphic.

1.1. **Notation.** We use 2^ω to denote the set of infinite binary sequences, $2^{<\omega}$ for the finite binary sequences and 2^n for the binary sequences of length n. A *tree* S is a downward closed subset of $2^{<\omega}$. The set of paths through S is denoted by $[S] \subseteq 2^\omega$. If S is a tree, $S_\tau = \{\sigma \in S, \sigma \subseteq \tau \vee \tau \subseteq \sigma\}$. Given $s \in \omega$, let $S[s] = S \cap 2^{\leq s}$ and $[S][s] = S \cap 2^s$, the stage-s approximation to S and $[S]$. A string $\tau \in S$ is *dead at stage s* if it has no extensions in $[S][s]$. We abuse notation and use $[\tau]$ represent both $\{X \in 2^\omega : \tau \subset X\} \subseteq 2^\omega$ and also $(2^{<\omega})_\tau \subseteq 2^{<\omega}$; it should be clear from the context which ones is being used. The empty string is denoted by $\emptyset$, concatenation of strings by $\sigma^\frown\tau$, and σ^- is the string σ with the last element removed. In general, when we use a variable, say x during a construction, $x[s]$ represents the value of x at stage s. If $x[s]$ is not specifically given a value, then it keeps the value of $x[s-1]$. Other notation will be as in Soare [22].

2. Perfect and thin versus finite

Theorem 2.1. *There is a uniform procedure which, given a computable tree $T \subseteq 2^{<\omega}$, builds a computable tree S such that, if $[T]$ is perfect $[S]$ is perfect and thin, and if $[T]$ is not perfect $[S]$ is finite.*

This theorem will be used in both constructions, the one of a non-slender thin class, and the one of a slender thin class. The proof starts developing ideas the will be used in both of those constructions.

In the case when $[T]$ is perfect, we will define two tree-embeddings $f, r \colon 2^{<\omega} \to S$ satisfying that $f(\emptyset) = \emptyset$ and for every $\sigma \in 2^{<\omega}$,

(fr1) $f(\sigma) \subseteq r(\sigma)$, $f(\sigma^\frown 0) = r(\sigma)^\frown 0, f(\sigma^\frown 1) = r(\sigma)^\frown 1$, and
(fr2) $[S_{f(\sigma)}] = [S_{r(\sigma)}]$

One can then prove by induction on n that $[S] = \bigcup_{\sigma \in 2^n} [S_{r(\sigma)}] = \bigcup_{\sigma \in 2^{n+1}} [S_{f(\sigma)}]$. It follows that $[S] = [image(f)] = [image(r)]$, and hence that $[S]$ is perfect.

There are two types of requirements: the *thinness requirements*

$$\mathcal{T}_e : \quad F_e \subseteq S \Rightarrow \exists C \subseteq 2^\omega \text{ clopen } ([F_e] = [S] \cap C),$$

where $\{F_0, F_1, F_2 ...\}$ is a sequence of computable subtrees of S enumerating all the Π^0_1 subclasses of $[S]$; and the *isolation requirements*

$$\mathcal{F}_e : \quad [T_{t_e}] \text{ is isolated} \Rightarrow [S] \text{ is finite.}$$

where $\{t_0, t_1, ...\}$ is an enumeration of T. These requirements are subdivided even further. Each thinness requirement $\mathcal{T}_e$ is divided into 2^{2e} sub-requirements $\mathcal{T}_\sigma$, one for each $\sigma \in 2^{2e}$.

$$\mathcal{T}_\sigma : \quad \text{either } [S_{f(\sigma)}] = [(F_e)_{f(\sigma)}], \text{ or } [F_e] \cap [S_{f(\sigma)}] = \emptyset.$$

Note that if all the requirements $\mathcal{T}_\sigma$ for $\sigma \in 2^{2e}$ are satisfied, then so is $\mathcal{T}_e$ by letting $C = \bigcup\{[f(\sigma)] : \sigma \in 2^{2e} \;\&\; [S_{f(\sigma)}] = [(F_e)_{f(\sigma)}]\}$. The strategy of $\mathcal{T}_\sigma$ is roughly the following. If $\mathcal{T}_\sigma$ sees the opportunity to define $r(\sigma) \notin F_e$, it will do it guaranteeing that $[F_e] \cap [S_{r(\sigma)}] = \emptyset$. If such an opportunity never appears, it will be because $(F_e)_{f(\sigma)} = S_{f(\sigma)}$.

Each isolation requirement $\mathcal{F}_e$ is be divided into 2^{2e+1} sub-requirements $\mathcal{F}_\sigma$, one for each $\sigma \in 2^{2e+1}$.

$\mathcal{F}_\sigma$: if $[T_{t_e}]$ is isolated $\Rightarrow [S_{f(\sigma)}]$ is isolated.

$\mathcal{F}_\sigma$ works roughly as follows. Every time it believes $[T_{t_e}]$ is isolated, it will kill all the paths in $[S_{f(\sigma)}][s]$, except for one. If this occurs infinitely often it is because $[T_{t_e}]$ is isolated and it will make $[S_{f(\sigma)}]$ isolated too. It then follows that $[S]$ has at most 2^{2e+1} many paths. Otherwise, after some stage $\mathcal{F}_\sigma$ will not act anymore and let the construction above $f(\sigma)$ continue.

2.1. **Organization of the construction.** We will define a computable tree S by stages; at stage s we will define $S[s] = S \cap 2^{\leq s}$. The functions f and r are also defined by stages and their values might change along the construction. At the end of stage s, we will have $f[s]$ and $r[s]$ defined on a finite tree $D_s \subset 2^{<\omega}$. We will always have that, if $\tilde{D}_s$ is the set of end-nodes of D_s, then $[S][s] = \{r(d)[s] : d \in \tilde{D}_s\} \subseteq 2^s$.

Each $\sigma \in 2^{<\omega}$ has a requirement $\mathcal{R}_\sigma$ (either $\mathcal{T}_\sigma$ or $\mathcal{F}_\sigma$) assigned. If $\sigma, \tau \in 2^{<\omega}$ are incomparable, then the requirements $\mathcal{R}_\sigma$ and $\mathcal{R}_\tau$ do not interact at all with each other, and none of the two requirements has stronger priority than the other one. If $\sigma \subset \tau$, then $\mathcal{R}_\sigma$ has stronger priority than $\mathcal{R}_\tau$ and it is allowed to cancel it. Cancellation of $\mathcal{R}_\tau$ by $\mathcal{R}_\sigma$ is all the interaction there is between $\mathcal{R}_\sigma$ and $\mathcal{R}_\tau$. Requirement $\mathcal{R}_\sigma$ is given $f(\sigma)$, and is responsible for defining $r(\sigma)$, extending $f(\sigma)$, and satisfying condition (fr2).

Main module of the Construction. At each stage s, we will start by *activating* the strategy for $\mathcal{R}_\emptyset$. This strategy might later activate $\mathcal{R}_{\langle 0\rangle}$ and then $\mathcal{R}_{\langle 1\rangle}$. Then, $\mathcal{R}_{\langle 0\rangle}$ could activate $\mathcal{R}_{\langle 0,0\rangle}$ and $\mathcal{R}_{\langle 0,1\rangle}$ and so on. D_s is the set to the requirements that are activated at stage s. In general, when a requirement $\mathcal{R}_\sigma$ is activated, at at stage $s+1$, it can do three things:

- The first time $\mathcal{R}_\sigma$ is active (either first time ever or first time since it was last canceled), it has to be *initialized.* $\mathcal{R}_\sigma$ defines $f(\sigma)$ using (fr1): If $\sigma = \tau^\frown i$, then $f(\sigma)[s+1] = r(\tau)[s+1]^\frown i$. (If $\sigma = \emptyset$, let $f(\sigma) = \emptyset$.) It defines $r(\sigma)[s+1] = f(\sigma)[s+1]$. It also set its status to an initial status that depends on the requirement. $\mathcal{R}_\sigma$ will not activate any other requirement at this stage, and hence we will have $\sigma \in \tilde{D}_{s+1}$. We will observe later that $r(\sigma)[s+1]$ has length $s+1$, because $r(\tau)[s+1]$ had to have length s.
- $\mathcal{R}_\sigma$ might *act.* In this case, $\mathcal{R}_\sigma$ will redefine $r(\sigma)$, cancel all the requirements of lower priority and stop going up the tree. So, again we will have $\sigma \in \tilde{D}_{s+1}$. $\mathcal{R}_\sigma$ is also responsible for defining $[S_{f(\sigma)}][s+1] \subseteq 2^{s+1}$.
- Otherwise, $\mathcal{R}_\sigma$ keeps the previous value of $r(\sigma)$ and activates $\mathcal{R}_{\sigma^\frown 0}$ and $\mathcal{R}_{\sigma^\frown 1}$. In this case we will have $\sigma \in D_s$ but $\sigma \notin \tilde{D}_{s+1}$. Since there is no interaction between $\mathcal{R}_{\sigma^\frown 0}$ and $\mathcal{R}_{\sigma^\frown 1}$, it does not matter whether they run simultaneously or one after the other one.

$\Diamond$

In the case when $[T]$ is perfect, the construction will be a finite injury one. Every requirement will be activated infinitely often, but it will stop canceling weaker priority ones after some stage and $f[s]$ and $r[s]$ will reach a limit. When $[T]$ is not perfect, and t_e is the first node such that $[T_{t_e}]$ is isolated, every requirement $\mathcal{R}_\sigma$ with $|\sigma| > 2e+1$ will be canceled infinitely often. However, in this case, requirement $\mathcal{F}_e$ is the only one that needs to be satisfied.

We observe that when $\mathcal{R}_\sigma$ is initialized, necessarily $f(\sigma)[s+1] = r(\sigma)[s+1]$ have length $s+1$: Since $\mathcal{R}_\sigma$ became activated at this stage, it means that for every $\gamma \subset \sigma$, $\mathcal{R}_\gamma$ did not act, and hence $r(\gamma)[s] = r(\gamma)[s+1]$. Another observation is that since $\mathcal{R}_\sigma$ was not active at stage s, it means that $\sigma \notin D_s$, but $\tau = \sigma^-$ had been initialized before s, so $\tau \in D_s$. Hence $r(\tau)[s] = r(\tau)[s+1]$ has length s, and $f(\sigma)[s+1] = r(\sigma)[s+1]$ has length $s+1$. The value of $f(\sigma)$ will not change again, unless $\mathcal{R}_\sigma$ is canceled by a stronger priority requirement. The value of $r(\sigma)$ might change a few times before stabilizing. Every time $r(\sigma)$ changes, $\mathcal{R}_\sigma$ initializes all the weaker priority requirements, that is, all the $\mathcal{R}_\tau$ with $\tau \supset \sigma$.

We now describe the strategies of the requirements $\mathcal{T}_\sigma$ and $\mathcal{F}_\sigma$.

2.2. **Thinness requirement.** Consider $\sigma \in 2^{<\omega}$, $|\sigma| = 2e$. Recall that $\mathcal{T}_\sigma$ is the requirement: either $[(F_e)_{f(\sigma)}] = [S_{f(\sigma)}]$, or $[F_e] \cap [S_{f(\sigma)}] = \emptyset$.

Module for requirement $\mathcal{T}_\sigma$. Suppose we are at stage $s+1$ and $\mathcal{T}_\sigma$ has just been activated. Also assume that $\mathcal{T}_\sigma$ has been initialized in some previous stage. So $f(\sigma)$ and $r(\sigma)$ have been previously defined, and $\mathcal{T}_\sigma$ is in status either `wai` (for "waiting") or `sat` (for satisfied). The initial status is `wai`.

First let us assume the current status of $\mathcal{T}_\sigma$ is `wai`. Check whether $[(F_e)_{f(\sigma)}][s] = [S_{f(\sigma)}][s]$.

- If so, we keep the status `wai` and activate requirements $\mathcal{F}_{\sigma^\frown 0}$ and $\mathcal{F}_{\sigma^\frown 1}$.
- Otherwise we *act*. Consider $\gamma \in [S_{f(\sigma)}] \setminus [F_e][s]$. We let $r(\sigma) = \gamma^\frown 0$ and $[S_{f(\sigma)}][s+1] = \{r(\sigma)\}$, making sure that $[F_e] \cap [S_{f(\sigma)}] = [F_e] \cap [S_{r(\sigma)}] = \emptyset$. The status of $\mathcal{T}_\sigma$ is set to `sat`. All the requirements $\mathcal{R}_\tau$ for $\tau \supset \sigma$ are canceled.

If $\mathcal{T}_\sigma$ is in status `sat` when it is activated, it immediately passes control to $\mathcal{F}_{\sigma^\frown 0}$ and $\mathcal{F}_{\sigma^\frown 1}$. $\Diamond$

Suppose there is a stage s_0 after which $\mathcal{T}_\sigma$ is activated infinitely often and never canceled again. If at some stage $s \geq s_0$, $\mathcal{T}_\sigma$ acts, then it is satisfied for ever, it status will be `sat` from there on, and after stage s, it will never act and cancel lower priority requirements again. Otherwise, $\mathcal{T}_\sigma$ never acts after s_0 and its status is always `wai`. In this case we have to have that $[(F_e)_{f(\sigma)}] = [S_{f(\sigma)}]$, so $\mathcal{T}_\sigma$ is also satisfied.

2.3. **Isolation requirements.** Consider $\sigma \in 2^{<\omega}$, $|\sigma| = 2e+1$. Recall that $\mathcal{F}_\sigma$ is the requirement: if $[T_{t_e}]$ is isolated, $[S_{f(\sigma)}]$ is isolated, where $t_e \in T$. We say that $n > |t_e|$ is *verified at s* if exactly one string $\tau \in T_{t_e} \cap 2^n$ is not dead at stage s. So, we have that $[T_{t_e}]$ is isolated if and only if for every $n > |t_e|$, there exists a stage s at which n is verified. The strategy for $\mathcal{F}_e$ is to try to verify every $n > |t_e|$ one by one. At each stage there is a number n_σ that we are waiting to be verified; once it is verified, we add one to the value of n_σ.

Module for requirement $\mathcal{F}_\sigma$. Suppose that we are at stage $s+1$. If $\mathcal{F}_\sigma$ has to be initialized, it sets its initial status to `niso` for not isolated (this is not relevant in this proof), and sets $n_\sigma[s+1] = |t_e| + 1$. At later stages, there will be some other value of $n_\sigma > |t_e|$, which is waiting to be verified. Suppose now that $\mathcal{F}_\sigma$ has been activated at stage $s+1$ and that it has already been initialized at some previous stage. Start by checking whether $n_\sigma[s]$ gets verified as s.

- If so, we momentarily believe that $[T_{t_e}]$ is isolated and we *act*. We define $S_{f(\sigma)}[s+1]$ so that only one string in $[S_{f(\sigma)}][s]$ is extended to $[S_{f(\sigma)}][s+1]$ and we let $r(\sigma)[s+1]$ be that one extension. We let $n_\sigma[s+1] = n_\sigma[s]+1$. We then cancel all the requirement $\mathcal{R}_\tau$ for $\tau \supset \sigma$ and we stop going up $2^{<\omega}$ for this stage. The status of $\mathcal{F}_\sigma$ is set to `iso` for isolated.
- Otherwise, we pass control to $\mathcal{T}_{\sigma^\frown 0}$ and $\mathcal{T}_{\sigma^\frown 1}$. The status of $\mathcal{F}_\sigma$ is set to `niso` for not isolated.

(We mentioned the status of $\mathcal{F}_\sigma$ only because we will use it in the Section 4) $\Diamond$

Suppose there is a stage s_0 after which $\mathcal{T}_\sigma$ is activated infinitely often and never canceled again. Note that if $[T]$ is perfect, then for every t_e, there will be some n_e which will never be verified. After that n_e is chosen by $\mathcal{F}_\sigma$, $\mathcal{F}_\sigma$ will never act again and let the lower priority requirement do their work. Also, if $[T_{t_e}]$ is empty, there will also be some n_e which will never be verified. On the other hand, if $[T]$ is neither perfect nor empty, for some t_e, $[T_{t_e}]$ is isolated. For each requirement $\mathcal{F}_\sigma$, $\sigma \in 2^{2e+1}$, every $n > |t_e|$ will be verified at some stage, and hence there be infinitely many stages with $[S_{f(\sigma)}][s]$ having only one element. So, $[S_{f(\sigma)}]$ will consist of an isolated path. $\mathcal{F}_\sigma$ will keep on injuring the requirements $\mathcal{R}_\tau$ for $\tau \supset \sigma$. But, since we are assuming that $S_{f(\sigma)}$ is isolated, we do not need to worry about them.

2.4. **Verifications.**

Lemma 2.2. *Suppose that for every $i < e$, $[T_{t_i}]$ is not isolated. Consider $\sigma \in 2^{\leq 2e+1}$.*

1. *There is a stage s_0, after which $\mathcal{R}_\sigma$ is always activated. In other words, $\sigma \in D_s$ for every $s \geq s_0$. Also, $\mathcal{R}_\sigma$ is never canceled after s_0, and $f(\sigma) = \lim_s f(\sigma)[s]$ exists and equals $f(\sigma)[s_0]$.*
2. *$\mathcal{R}_\sigma$ is satisfied.*
3. *For $\sigma \in 2^{<2e+1}$, $\mathcal{R}_\sigma$ acts only finitely often, and $r(\sigma) = \lim_s r(\sigma)[s]$ exists.*

Proof. The proof is by simultaneous induction on the length of σ. Part (1) follows from the inductive hypothesis of (1) and (3). Parts (2) and (3) follow from (1) and the comments after the description above of the modules for the requirements. $\square$

Now, if $[T]$ is perfect, then the lemma above holds for every e, and hence all the requirements $\mathcal{T}_e$ are satisfied. So $[S]$ is a thin Π^0_1-class. Also, the functions f and r are defined everywhere and satisfy (fr1) and (fr2). So $[S]$ is perfect. Otherwise, $[T]$ has some isolated path. Let e be the least such that $[T_{t_e}]$ is isolated. From the lemma it follows that, for each $\sigma \in 2^{2e+1}$, $\mathcal{F}_\sigma$ is satisfied, and hence $\mathcal{F}_e$ is satisfied. So $[S]$ has at most 2^{2e+1} paths.

2.5. **A small modification.** We now describe a stronger version of Theorem 2.1 that we are going to need in the construction of a slender thin Π^0_1 class. The idea of the proof will also be used in that construction.

We still have a Π^0_1 class $[T]$ and we want to define S as in Theorem 2.1. But suppose now that not allowed to define $[S][s]$ at every stage s, but only at some infinite number of stages, and there is some foreign agent defining $S[s]$ at the other stages. However if we defined $[S][s]$ a certain way and the foreign agent is defining $[S][t]$ for some $t > s$, there has to be an extension in $[S][t]$ of every element of $[S][s]$. This way, he is not really killing our construction. Let us describe this in a more formal way.

Lemma 2.3. *Let $[T]$ be a Π^0_1 class. There is a computable function Γ_T which takes as input a finite sequence of stages $s_0 < s_1 < .... < s_n$ and a subtree of $2^{<s_n}$, and outputs a subtree of $2^{\leq s_n}$ extending it, and satisfies the following property. Consider any infinite computable sequence $\{s_0 < s_1 < ....\}$ and a computable tree $S \subseteq 2^{<\omega}$ such that for every $n \in \omega$, $S[s_n] = \Gamma_T(s_0, ..., s_n, S[s_n - 1])$ and every element of $[S][s_n]$ has an extension in $[S][s_{n+1} - 1]$. Then, if T is perfect, $[S]$ is perfect and thin, and $[S]$ is finite otherwise.*

Proof. Just let $\Gamma_T(s_0, ..., s_n, S[s_n-1])$ do what the construction of Theorem 2.1 does in one stage. Before, for each $\sigma \in \tilde{D}_{s_{n-1}}$ define $r(\sigma)[s_n-1]$ to be some extension of $r(\sigma)[s_n-1]$ in $S[s_n-1]$. It is not hard to see that this does not affect the satisfaction of the requirements $\mathcal{T}_\sigma$ and $\mathcal{F}_\sigma$. □

3. A thin, non-slender class

Theorem 3.1. *For every Δ^0_2 Boolean algebra $\mathcal{B}$ with infinitely many atoms, there exists a thin but not slender computable tree S whose lattice of Π^0_1 subclasses is isomorphic to $\mathcal{B}$.*

Definition 3.2. Given a set $X \subseteq 2^\omega$, the *algebra of clopen set of* X, $\mathrm{clo}(X)$ is the Boolean algebra whose elements are of the form $C \cap X$, where C is a clopen subset of 2^ω.

If T is a computable tree, we write $\mathrm{clo}(T)$ for $\mathrm{clo}([T])$.

Note that if $[T] \subseteq 2^\omega$ is a thin Π^0_1 class, then the lattice of Π^0_1 subclasses of $[T]$ coincides with the algebra of clopen sets of $[T]$.

Also observe that $\mathrm{clo}(T)$ is isomorphic to the Boolean algebra of clopen sets of 2^ω modulo the equivalence relation $C \equiv D \iff C \cap [T] = D \cap [T]$, which is a Δ^0_2-condition, and that the elements of $\mathrm{clo}(2^\omega)$ can be represented by finite sets of binary strings. It follows that for a computable tree T, $clo(T)$ is Δ^0_2 presentable.

Lemma 3.3. *For every Δ^0_2 Boolean algebra $\mathcal{B}$, there is a computable tree T whose algebra of clopen sets is $\mathcal{B}$.*

Proof. Feiner [12] proved that every Δ^0_2 Boolean algebra is isomorphic to a c.e. quotient Boolean algebra. Then, cited as folklore, it is proven in [3, Theorem 4.8] that every c.e. quotient Boolean algebra is of the form $clo(T)$ for some computable tree T. □

Fix such a computable tree T.

We build a computable tree S and two tree-embeddings $f, r \colon T \to S$ satisfying condition (fr1) for $\sigma \in T$, but not (fr2). Unfortunately, we will not have $[S] = [image(f)] = [image(r)]$ as in the previous construction. Instead, we will construct S, f and r with the following properties.

(Sfr1) If $[T_\sigma]$ is empty, then so is $[S_{f(\sigma)}]$;
(Sfr2) If $[T_\sigma]$ is isolated, then $[S_{f(\sigma)}]$ is finite;
(Sfr3) If $[T_\sigma]$ has more than one element and is not perfect , then $[S_{f(\sigma)}] = X_\sigma \cup [S_{r(\sigma)}]$, where $X_\sigma \subseteq 2^\omega$ is a finite set disjoint form $[r(\sigma)]$.
(Sfr4) If $[T_\sigma]$ is perfect, then $[S_{f(\sigma)}] = X_\sigma \cup [S_{r(\sigma)}]$, where $X_\sigma \subseteq 2^\omega$ is either perfect or empty, and is disjoint form $[r(\sigma)]$.

Using Remmel-Vaught's theorem, we can prove that these conditions imply that $\mathrm{clo}(S) \cong \mathrm{clo}(T) \cong \mathcal{B}$. If we also manage to make S thin, we will have that the lattice of Π^0_1 subclasses of $[S]$ is isomorphic to $\mathcal{B}$.

Theorem 3.4 (Remmel-Vaught [18])**.** *Let $\mathcal{B}_0$ and $\mathcal{B}_1$ be Boolean algebras with infinitely many atoms. Suppose $\varphi\colon \mathcal{B}_0 \to \mathcal{B}_1$ is a Boolean algebra embedding such that*

(1) *$\mathcal{B}_1$ is generated by the image of φ and the atoms of $\mathcal{B}_1$;*
(2) *every atom of $\mathcal{B}_0$ is mapped to a finite sum of atoms in $\mathcal{B}_1$; and*
(3) *every atom of $\mathcal{B}_1$ is below the image of an atom of $\mathcal{B}_0$.*

Then, $\mathcal{B}_0$ and $\mathcal{B}_1$ are isomorphic.

Lemma 3.5. *If S, f and r satisfy conditions (fr1), (Sfr1)-(Sfr4) above, then the clopen Boolean algebra of S is isomorphic to the one of T, namely $\mathcal{B}$.*

Proof. We define a map $\varphi\colon \mathrm{clo}(T) \to \mathrm{clo}(S)$ which satisfies the conditions in Vaught-Remmel's theorem. If $[S_{f(\sigma)}] = X_\sigma \cup [S_{r(\sigma)}]$, where X_σ is a finite set, by (Sfr3) we know that $[S_{r(\sigma)}]$ contains at least one isolated path; choose one and call it I_σ. The idea is to put $[S_{r(\sigma)}]$ together with I_σ below the image, under φ, of some atom of $\mathrm{clo}(T)$. We first define φ on T by recursion, and then extend it to $\mathrm{clo}(T)$ in the obvious way. We abuse notation, and when we write $\varphi(\sigma)$ for $\sigma \in T$, we actually mean $\varphi([\sigma] \cap [T])$. Let $\varphi(\emptyset) = \emptyset$. Now we want to define $\varphi(\tau)$. If $[T_\tau]$ is not perfect, define

$$\varphi(\tau) = f(\tau) \cup \bigcup_{\sigma \subseteq \tau : I_\sigma \in [T_\tau]} X_\sigma$$

If $[T_\tau]$ is perfect, then, by (Sfr4), so is $[S_{f(\tau)}]$. Define φ mapping $\mathrm{clo}(T_\tau)$ to $\mathrm{clo}(S_{f(\tau)})$ isomorphically.

It is not hard to see that φ can be extended to a Boolean algebra embedding. All one needs to check is that for every $\tau \in T$, $\varphi(\tau) = \varphi(\tau^\frown 0) \cup \varphi(\tau^\frown 1)$ and that $\varphi(\tau^\frown 0) \cap \varphi(\tau^\frown 1) = 0$. It is also not hard to see that every $\sigma \in S$ with $[S_\sigma]$ perfect is in the image of φ. If $[S_\sigma]$ is not perfect, then $[S_\sigma]$ together with some finite set is in the image of φ. In the case $[S_\sigma]$ is isolated and contained in some X_π which is finite, we have that $[S_\sigma]$ is below the image of $I_\pi \in [T]$. So, the conditions in Remmel-Vaught's theorem are satisfied. $\square$

There are three types of requirements: the thinness requirements $\mathcal{T}_\sigma$, one for every $\sigma \in T$, σ of length $3e$; the isolation requirements $\mathcal{F}_\sigma$, one for every $\sigma \in T$, σ of length $3e+1$; and the non-slenderness requirements $\mathcal{N}_\sigma$, one for every $\sigma \in T$, σ of length $3e+2$. The construction of S, f, and r is again a finite injury one, and it goes exactly as the *Main module of the construction* in Section 2.1. There is one slight difference. Only the nodes $\sigma \in T$ have requirements assigned. Some nodes $\sigma \in T$ have no extensions in $[T]$. These are the ones that we call *dead* nodes. But it might take us a while to find this out, and requirement $\mathcal{R}_\sigma$ will start its job as usual. After the stage s when we find out that σ is a dead node (i.e. σ has no extensions in $T[s]$), we do not need to work for $\mathcal{R}_\sigma$ anymore. We could do a bit more work, if we actually want f and r to be defined at every node of T. The next time $\mathcal{R}_\sigma$ becomes activated, we do nothing, and we stop building S above $f(\sigma)$. This way we satisfy (Sfr1)

The thinness and the isolation requirements work exactly as the modules described for the previous construction. Let us now describe how the non-slenderness requirements work. These requirements do not injure lower priority requirements.

3.1. **Non-slenderness requirements.** The non-slenderness requirement $\mathcal{N}$ will construct a computable tree $F \supseteq S$ such that for no clopen set C we have $\mathrm{int}([F]) \cap \mathrm{iso}([S]) = C \cap \mathrm{iso}([S])$. $\mathcal{N}$ has infinitely many sub-requirements $\mathcal{N}_\sigma$, one for each $\sigma \in T$, $|\sigma|$ of the form $3e+2$.

$$\begin{aligned} \mathcal{N}_\sigma: \quad & [T_\sigma] \text{ has an isolated path} \Rightarrow \exists \text{ finite sets } Z_\sigma, Y_\sigma \subset 2^\omega \text{ such that} \\ & [S_{f(\sigma)}] = Z_\sigma \cup Y_\sigma \cup [S_{r(\sigma)}] \quad \& \quad Z_\sigma \subset \mathrm{int}([F]) \quad \& \quad Y_\sigma \cap \mathrm{int}([F]) = \emptyset, \end{aligned}$$

We claim that if all the requirements $\mathcal{N}_\sigma$ are satisfied, then S is not slender. We know that $[T]$ has infinitely many isolated paths. Let $\{\sigma_0, \sigma_1, ...\} \subseteq T$ an enumeration of the roots of all this isolated paths (that is, for each σ_i, $[T_{\sigma_i}]$ is an isolated path, but $[T_{\sigma_i^-}]$ is not). For each i, let $\tau_i \subseteq \sigma_i$ be the longest string whose length is of the form $3e+2$. Since we are assuming $\mathcal{N}_{\tau_i}$ is satisfied, we have two finite sets $Z_{\tau_i}, Y_{\tau_i} \subset \mathrm{iso}[S] \cap [f(\tau_i)]$, such that $Z_\sigma \subset \mathrm{int}([F])$ and $Y_\sigma \cap \mathrm{int}([F]) = \emptyset$. Suppose toward a contradiction there is a clopen set C such that $\mathrm{int}([F]) \cap \mathrm{iso}([S]) = C \cap \mathrm{iso}([S])$, and suppose that C is a finite union of basic open sets $[\pi_j]$, with $|\pi_j| < k$. Let i be such that $|\tau_i| > k$ and hence $|f(\tau_i)| > k$. Then, either $[f(\tau_i)] \subseteq C$ or $[f(\tau_i)] \cap C = \emptyset$, contradicting $Z_{\tau_i} \subseteq C$ and $Y_{\tau_i} \cap C = \emptyset$.

Module for $\mathcal{N}_\sigma$ at stage $s+1$. Suppose first that $\mathcal{N}_\sigma$ has been initialized at the previous stage, namely stage s. So, no requirement of lower priority has been initialized yet. Defines $r(\sigma) = f(\sigma)^\frown 0$; this value will not change unless $\mathcal{N}_\sigma$ is canceled. (Note that since $\mathcal{N}_\sigma$ was initialized at stage s, $|f(\sigma)| = s$.) $\mathcal{N}_\sigma$ also enumerates $f(\sigma)^\frown 0$ and $f(\sigma)^\frown 1$ into S and hence into F too. This is all it does at this stage, and it does not activate any other requirement.

In the next stages it will build Z_σ and Y_σ extending $f(\sigma)^\frown 10$ and $f(\sigma)^\frown 11$ respectively, and will let the rest of the construction continue in top of $f(\sigma)^\frown 0$. Here is how it builds Z_σ and Y_σ. Let Q_σ be the tree obtained when Theorem 2.1 is applied to T_σ. So, $[Q_\sigma]$ is perfect and thin if T_σ is perfect, and $[Q_\sigma]$ is finite otherwise. Requirement $\mathcal{N}_\sigma$ places two copies of Q_σ in S and F, one in top of $f(\sigma)^\frown 10$ and one in top of $f(\sigma)^\frown 11$. Since it might be cancel later, $\mathcal{N}_\sigma$ builds these extensions step by step. Recall that at a stage $s+1$ we can only define S up to length $s+1$. $\mathcal{N}_\sigma$ will include the whole cone $[f(\sigma)^\frown 10]$ inside $F_{f(\sigma)^\frown 10}$, but it will let $F_{f(\sigma)^\frown 11} = S_{f(\sigma)^\frown 11} = f(\sigma)^\frown 11^\frown Q_\sigma$.

So, the actions taken at stage $s+1$, assuming $\mathcal{N}_\sigma$ was initialized at a previous stage $t = |f(\sigma)| < s$, are the following. Keep $r(\sigma) = f(\sigma)^\frown 0$; Define $[S_{f(\sigma)^\frown 10}][s+1] = f(\sigma)^\frown 10^\frown [Q_\tau][s-t-1]$, $F_{f(\sigma)^\frown 10}[s+1] = [f(\sigma)^\frown 10][s+1]$ and $[F_{f(\sigma)^\frown 11}][s+1] = [S_{f(\sigma)^\frown 11}][s+1] = f(\sigma)^\frown 11^\frown [Q_\tau][s-t-1]$; Activate requirements $\mathcal{T}_{\sigma^\frown 0}$ and $\mathcal{T}_{\sigma^\frown 1}$. $\Diamond$

Suppose there is a stage s_0 after which $\mathcal{N}_\sigma$ is always activated and never canceled again. It is clear that $\mathcal{N}_\sigma$ will manage to build Y_σ and Z_σ as desired, and satisfy (Sfr3) and (Sfr4). All the requirements of lower priority than $\mathcal{N}_\sigma$ are initialized after s_0+1 and never canceled by $\mathcal{N}_\sigma$.

3.2. **Thinness requirements.** For each $\sigma \in T$, σ of length $3e$ we have a thinness requirement: either $[(F_e)_{f(\sigma)}] = [S_{f(\sigma)}]$, or $[F_e] \cap [S_{f(\sigma)}] = \emptyset$. It does exactly the same as the module described in Section 2.2. However, when we showed that the satisfaction of all the $\mathcal{T}_\sigma$ for $\sigma \in 2^{2e+1}$ implies the satisfaction of $\mathcal{T}_e$ in the previous construction, we used the fact that $[S] = \bigcup_{\sigma \in 2^{2e+1}} [S_{f(\sigma)}]$, which is not true in this case. Now, by (Sfr3) and (Sfr4), we have that

$$[S] = \bigcup_{\sigma \in T \cap 2^{3e+1}} [S_{f(\sigma)}] \cup \bigcup_{\tau \in T \cap 2^{<3e+1}} X_\tau.$$

This is not a problem, because for each $\tau \in T \cap 2^{<3e+1}$, X_τ is thin.

3.3. **Isolation requirements.** For each $\sigma \in T$, σ of length $3e+1$ we have a isolation requirement: if $[T_\sigma]$ is isolated, $[S_{f(\sigma)}]$ is isolated . It does exactly the same as the module described in the previous construction, except that now we are looking at whether $[T_\sigma]$ is isolated instead of $[T_{t_e}]$. The objective of these requirements is to satisfy (Sfr2): Suppose $[T_\tau]$ is isolated and $\sigma \supseteq \tau$ is the initial segment of $[T_\tau]$ of length $3e+1$. Then, by (Sfr1) and (Sfr3), if $\mathcal{N}_\sigma$ is satisfied, $[S_{f(\tau)}]$ is finite, and hence (Sfr2) is satisfied.

3.4. **Verifications.**

Lemma 3.6. *Suppose that for every $\sigma \subset \tau \in T$, $[T_\sigma]$ is neither empty nor isolated. Consider $\sigma \subseteq \tau$.*

(1) *There is a stage s_0, after which $\mathcal{R}_\sigma$ is always activated. In other words, $\sigma \in D_s$ for every $s \geq s_0$. Also, $\mathcal{R}_\sigma$ is never canceled after s_0, and $f(\sigma) = \lim_s f(\sigma)[s]$ exists and equals $f(\sigma)[s_0]$.*
(2) *$\mathcal{R}_\sigma$ is satisfied.*
(3) *For $\sigma \subset \tau$, $\mathcal{R}_\sigma$ acts only finitely often, and $r(\sigma) = \lim_s r(\sigma)[s]$ exists.*

Proof. The proof is by simultaneous induction on the length of σ. □

It follows that S, f and r are as desired.

4. A thin, slender class

Theorem 4.1. *For every Δ^0_2 Boolean algebra $\mathcal{B}$ there exists a thin and slender computable tree S whose lattice of Π^0_1 subclasses is $\mathcal{B}$.*

Fix a computable T as the one given by Lemma 3.3.

This construction has three types of requirements: thinness requirements, isolation requirements and slenderness requirements. Each node $\sigma \in T$ will have a requirement assigned $\mathcal{R}_\sigma$ that can be of any of these three kinds as in the previous constructions, and the thinness and isolation requirements will work exactly as before. One difference with the previous constructions is that this is an infinite injury construction, because the slenderness requirements will have Π^0_2 and Σ^0_2 outcomes. The construction is organized on a tree of strategies; actually a tree of trees of strategies. So each $\mathcal{R}_\sigma$ will have a belief on the outputs of stronger priority requirements, namely $\{R_\tau : \tau \subset \sigma\}$. We will have different versions of $\mathcal{R}_\sigma$ for the different possible believes, as one usually has in tree-of-strategies arguments, and one of this versions will act infinitely often and get injured only finitely often.

We will construct a computable tree S by stages and functions f and r satisfying (fr1), (Sfr1)-(Sfr4). We start by describing how the Slenderness requirements work, and then we will explain how the construction is organized on the tree of strategies.

4.1. **Slenderness requirement.** For every computable tree F_e we have a *Slenderness requirement*:

$$\mathcal{S}_e: \quad \exists \text{ clopen } C \subseteq 2^\omega \quad (\text{int}[F_e] \cap \text{iso}[S] = C \cap \text{iso}[S]).$$

We partition this requirements into at most 2^{e+2} many requirements, one for each string $\sigma \in T$ of length $3e+2$:

$$\mathcal{S}_\sigma: \quad \text{either } \text{int}([F_e]) \cap [S_{f(\sigma)}] = \emptyset, \text{ or } [S_{r(\sigma)}] \subseteq \text{int}(F_e).$$

Note that if $\mathcal{S}_\sigma$ is satisfied for every string $\sigma \in T$ of length $3e+2$, then $\mathcal{S}_e$ is satisfied: Let $C_0 = \bigcup\{[r(\sigma)] : \sigma \in T \cap 2^{3e+2}, [S_{r(\sigma)}] \subseteq \text{int}[F_e]\}$. Then $\text{int}[F_e] \cap \text{iso}[S] = (C_0 \cap \text{iso}[S]) \cup X$, where $X = \text{int}[F_e] \cap \bigcup_{\tau \in T \cap 2^{<3e+2}} X_\tau$. Note that (Sfr1)-(Sfr4) imply that X contains a finite number of isolated paths. Let C be the union of C_0 and those isolated paths.

We say that $\tau \in S$ is *e-verified* at stage s if $\exists \gamma \in 2^{\leq s}(\gamma \supseteq \tau \ \& \ \gamma \notin F_e)$. So, we have that $\text{int}([F_e]) \cap [S_{f(\sigma)}] = \emptyset$ if and only if for every $\tau \in S_{f(\sigma)}$, there is a stage s at which either τ is e-verified or τ has no extension in $[S][s]$. $\mathcal{N}_\sigma$ will try to make sure that every $\tau \in S_{f(\sigma)}$ is either e-verified, and if so, we say it has outcome ∞. If instead we find a string $\tau \in S_{f(\sigma)}$ that is never e-verified, and hence $[\tau] \subseteq \text{int}([F_e])$, we will move the construction of $S_{r(\sigma)}$ to the cone above τ. In this case we say that $\mathcal{S}_\sigma$ has outcome `fin`. $\mathcal{S}_\sigma$'s initial status is ∞.

Suppose we are at stage $s+1$ and requirement $\mathcal{S}_\sigma$ gets activated.

- Suppose first that the last time we visited $\mathcal{S}_\sigma$ it had status ∞.

  If every $\tau \in S_{f(\sigma)}[s]$ is e-verified, then keep the status ∞ and move on to the next requirements $\mathcal{T}_{\sigma^\frown 0}$ and $\mathcal{T}_{\sigma^\frown 1}$.

  Otherwise, let τ_0 be the first $\tau \in S_{f(\sigma)}[s]$ that has not been e-verified (first in some ordering of $2^{<\omega}$). Let τ_1 be an extension of τ_0 in $[S][s]$. The plan now is to wait until τ_1 (actually $\tau_1{}^\frown 0$) is e-verified, which will imply that τ_0 is e-verified. While we wait, we move the construction of $[S_{f(\sigma)}]$ above τ_1. Define $r(\sigma)[s+1] = \tau_1{}^\frown 0$ (so it has length $s+1$) and set the status of $\mathcal{S}_\sigma$ at this stage to `fin`. If $\tau_1{}^\frown 0$ is never e-verified, then we will have $[S_{r(\sigma)}] \subseteq [r(\sigma)] \subseteq \text{int}(F_e)$ as wanted. If at some later stage it is e-verified, we will redefine $r(\sigma) = f(\sigma)$ and we forget we ever moved $r(\sigma)$ to $\tau_1{}^\frown 0$. The requirements of lower priority than σ will then continue the work they were doing when they were assuming $\mathcal{S}_\sigma$ had outcome ∞ and $r(\sigma) = f(\sigma)$. Therefore, for now, while $\mathcal{S}_\sigma$ has outcome `fin`, we cannot kill what we were doing above $f(\sigma)$ while we were believing that the the output of $\mathcal{S}_\sigma$ is ∞. So, at every stage $t > s$, while the outcome of $\mathcal{S}_\sigma$ is still `fin`, we have to make sure that every node in $[S][s]$ has at least one extension in $[S][t]$. We have to be careful doing this, because if we never come back to outcome ∞, we will had built some new paths extending $f(\sigma)$ but not $r(\sigma)$. Let $X_\sigma = [S_{f(\sigma)}] \setminus [S_{r(\sigma)}]$. So, we have to make X_σ satisfy conditions (Sfr1)-(Sfr4). We do it as in the previous construction: On top of each $\tau \in [S_{f(\sigma)}][s]$, $\tau \neq \tau_1$, we use Lemma 2.3 to build a tree Q_τ such that if $[T_\sigma]$ is perfect, then $[Q_\tau]$ is perfect and thin, and $[Q_\tau]$ is finite otherwise. Of course, the construction of Q_τ is done step by step every time $\mathcal{S}_\sigma$ is active and while it has outcome `fin`.
- Suppose now that the last time $\mathcal{S}_\sigma$ was active, it had status `fin`. That means that we are waiting for $r(\sigma)$ to get e-verified.

– If $r(\sigma)$ is still not e-verified, we keep the status $\mathtt{fin}$ and we activate the next requirements $\mathcal{T}_{\sigma^\frown 0}$ and $\mathcal{T}_{\sigma^\frown 1}$. We also do one more step in the construction each of the Q_τ that we started the last time we changed $\mathcal{S}_\sigma$'s status from ∞ to $\mathtt{fin}$. One thing to notice here is that even if $\mathcal{S}_\sigma$ stays in status $\mathtt{fin}$ for ever, it might not be active at every stage. The reason is that there might be some stronger requirement $\mathcal{S}_\pi$, $\pi \subset \sigma$, that has outcome ∞. Even though we are assuming $\mathcal{S}_\sigma$ knows this is $\mathcal{S}_\pi$'s final outcome, $\mathcal{S}_\pi$ is going to change its status infinitely often to $\mathtt{fin}$. Every time it does it, $\mathcal{S}_\sigma$ gets paralyzed, and when $\mathcal{S}_\pi$'s status comes back to ∞ and $\mathcal{S}_\sigma$ becomes active again, it will find that some of the paths it was constructing had been extended to longer paths, though $\mathcal{S}_\pi$ made sure no path had been killed. This does not affect $\mathcal{S}_\sigma$ at all. So long as $\mathcal{S}_\sigma$ gets to do a new step in the construction of the trees Q_τ infinitely often, it will manage to construct them satisfying (Sfr1)-(Sfr4).
– Suppose now that $r(\sigma)$ has been e-verified since the last time $\mathcal{S}_\sigma$ was active. Change $\mathcal{S}_\sigma$'s status to ∞. Define $r(\sigma)[s+1] = f(\sigma)$. Let s_0 be the last stage when $\mathcal{S}_\sigma$'s status was ∞. Each string in $[S_{f(\sigma)}][s_0]$ has at least one extension in $[S_{f(\sigma)}][s]$; choose one. Kill all the other stings in $[S_{f(\sigma)}][s]$ by not extending them in $[S_{f(\sigma)}][s+1]$. The rest of $[S_{f(\sigma)}][s+1]$ will be defined at the end of stage $s+1$ by other requirements. If a sting in $[S_{f(\sigma)}][s_0]$ was of the form $r(\pi)[s_0]$ for some $\pi \supset \sigma$, redefine $r(\pi)$ to be the chosen extension of it in $[S_{f(\sigma)}][s]$. Activate requirements $\mathcal{T}_{\sigma^\frown 0}$ and $\mathcal{T}_{\sigma^\frown 1}$.

4.2. **Organization of the construction.** Since this construction is an infinite injury one, we will do it on a tree of strategies. The way we do this is very standard, except for the fact that the requirements are not linearly ordered by priority. We could order the requirements linearly and define the tree of strategies the usual way, but instead we continue the style of the previous constructions.

Let TS, the *tree of strategies*, be the set of pairs $\langle\sigma,\alpha\rangle$ where $\sigma \in T$, and α contains beliefs of possible outcomes of the requirements stronger than $\mathcal{R}_\sigma$, that is, $\alpha \in \{\mathtt{wai},\mathtt{sat},\mathtt{iso},\mathtt{niso},\infty,\mathtt{fin}\}^{<\omega}$ satisfies that $\alpha(3e) \in \{\mathtt{sat},\mathtt{wai}\}$, $\alpha(3e+1) \in \{\mathtt{iso},\mathtt{niso}\}$, $\alpha(3e+2) \in \{\infty,\mathtt{fin}\}$, and $|\alpha| = |\sigma|$. So, for $i < |\sigma|$, we think of $\alpha(i)$ as the outcome of the requirement at $\sigma\restriction i$, and there is no belief about the outcome of σ. Each $\langle\sigma,\alpha\rangle \in TS$ has a requirement $\mathcal{R}_{\langle\sigma,\alpha\rangle}$ assigned, where $\mathcal{R}$ can be either $\mathcal{T}$, $\mathcal{F}$ or $\mathcal{S}$ depending on whether $|\sigma|$ is of the form $3e$, $3+1$ or $3e+2$. The outcomes are ordered by $\mathtt{sat} <_L \mathtt{wai}$, $\mathtt{iso} <_L \mathtt{niso}$ and $\infty <_L \mathtt{fin}$. This induces an ordering on TS as follows: we define $\langle\sigma_0,\alpha_0\rangle <_L \langle\sigma_1,\alpha_1\rangle$, and say that $\langle\sigma_0,\alpha_0\rangle$ is to the *left* of $\langle\sigma_1,\alpha_1\rangle$, if there exists a i such that $\sigma_0\restriction i+1 = \sigma_1\restriction i+1$, $\alpha_0\restriction i = \alpha_1\restriction i$ and $\alpha_0(i) <_L \alpha_1(i)$. We give $R_{\langle\sigma_0,\alpha_0\rangle}$ a *stronger priority* than $R_{\langle\sigma_1,\alpha_1\rangle}$ if either $\langle\sigma_0,\alpha_0\rangle <_L \langle\sigma_1,\alpha_1\rangle$ or $\langle\sigma_0,\alpha_0\rangle \subset \langle\sigma_1,\alpha_1\rangle$.

At every stage s there will be a finite tree $D_s \subset T$ of nodes that get visited and a function $o_s\colon D_s \to \{\mathtt{wai},\mathtt{sat},\mathtt{iso},\mathtt{niso},\infty,\mathtt{fin}\}$ of outcomes. For every $\sigma \in D_s$, the requirement $\mathcal{R}_{\langle\sigma,o_s\restriction\sigma\rangle}$ is activated at stage s, and has outcome, or status, $o_s(\sigma)$, where $o_s\restriction\sigma = \langle o_s(\sigma\restriction 0), o_s(\sigma\restriction 1), ..., o_s(\sigma\restriction|\sigma|-1)\rangle$.

At each stage s instead of having a partial function $f[s]\colon D_s \to S$, we have a partial function $f[s]\colon TS \to S$. The domain of $r[s]$ will be $TS+$ instead of TS, where $TS+$ is the set of pairs $\langle\sigma,\alpha\rangle$ where $\sigma \in T$, and α contains beliefs of possible outcomes of the requirements $\mathcal{R}_{\sigma\restriction i}$ for every $i \leq |\sigma|$, so $|\alpha| = |\sigma|+1$. $f[s]$ and $r[s]$ still satisfy that

$f(\langle\sigma^\frown 0,\alpha\rangle) = r(\langle\sigma,\alpha\rangle)^\frown 0$, $f(\langle\sigma^\frown 1,\alpha\rangle) = r(\langle\sigma,\alpha\rangle)^\frown 1$, and $f(\langle\sigma,\alpha^-\rangle) \subseteq r(\langle\sigma,\alpha\rangle)$. Each requirement $\mathcal{R}_{\langle\sigma,\alpha\rangle}$ is essentially given a string $f(\langle\sigma,\alpha\rangle)$ for it to start working. At each stage s, $\mathcal{R}_{\langle\sigma,\alpha\rangle}$ has an output $o_s(\sigma)$ and defines $r(\langle\sigma,\alpha^\frown o_s(\sigma)\rangle)[s]$ extending $f(\langle\sigma,\alpha\rangle)$.

Main module for the construction at stage s. We start by *activating* the strategy for $\mathcal{R}_{\langle\emptyset,\emptyset\rangle}$. In general, when a requirement $\mathcal{R}_{\langle\sigma,o_s\restriction\sigma\rangle}$ is activated, at at stage s, it will end up with some status; we let $o_s(\sigma)$ be that status, and we enumerate σ into D_s. All the requirements to the right of $\langle\sigma, o_s\restriction\sigma+\rangle$ are canceled, where $o_s\restriction\sigma+ = \langle o_s(\sigma\restriction i) : i = 0, ..., |\sigma|\rangle$. Depending on what type of action $\mathcal{R}_{\langle\sigma,o_s\restriction\sigma\rangle}$ takes, it might activate requirements $\mathcal{R}_{\langle\sigma^\frown 0,o_s\restriction\sigma+\rangle}$ and $\mathcal{R}_{\langle\sigma^\frown 1,o_s\restriction\sigma+\rangle}$. If it does not, $\mathcal{R}_{\langle\sigma,o_s\restriction\sigma\rangle}$ has to define $[S_{f(\langle\sigma,o_s\restriction\sigma\rangle)}][s]$. $\Diamond$

The *true path* $\mathtt{TP}\colon T \to \{\mathtt{wai}, \mathtt{sat}, \mathtt{iso}, \mathtt{niso}, \infty, \mathtt{fin}\}$ is defined as usual:

$$\mathtt{TP}(\sigma) = \liminf_{s:(\mathtt{TP}\restriction\sigma)=(o_s\restriction\sigma)} o_s(\sigma).$$

At the end of the construction we define, for $\sigma \in T$,

$$f(\sigma) = \lim_{s:(\mathtt{TP}\restriction\sigma)=(o_s\restriction\sigma)} f(\langle\sigma, \mathtt{TP}\restriction\sigma\rangle)[s]$$

and

$$r(\sigma) = \lim_{s:(\mathtt{TP}\restriction\sigma+)=(o_s\restriction\sigma+)} r(\langle\sigma, \mathtt{TP}\restriction\sigma+\rangle)[s].$$

4.3. **Modulo for the slenderness requirements.**

Modulo for requirement $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ at stage $s+1$.

- Suppose first that the last time we visited $\mathcal{S}_\langle\sigma,\alpha\rangle$ it had status ∞.
  - If every $\tau \in S_{f(\langle\sigma,\alpha\rangle)}[s]$ is e-verified, then let $o_{s+1}(\sigma) = \infty$ and move on to the next requirements $\mathcal{T}_{\langle\sigma^\frown 0,\alpha^\frown\infty\rangle}$ and $\mathcal{T}_{\langle\sigma^\frown 1,\alpha^\frown\infty\rangle}$.
  - Otherwise, let τ_0 be the first $\tau \in S_{f(\langle\sigma,\alpha\rangle)}[s]$ that has not been e-verified (first in some ordering of $2^{<\omega}$). Let τ_1 be an extension of τ_0 in $[S][s]$. Define $r(\langle\sigma,\alpha^\frown\mathtt{fin}\rangle)[s+1] = \tau_1{}^\frown 0$ and set $o_{s+1}(\sigma) = \mathtt{fin}$. Let $[S_{f(\langle\sigma,\alpha\rangle)}][s+1] = \{\tau^\frown 0 : \tau \in [S_{f(\langle\sigma,\alpha\rangle)}][s]\}$. Do not activate any other requirements.
- Suppose now that the last time $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ was active, it had status $\mathtt{fin}$.
  - If $r(\langle\sigma,\alpha^\frown\mathtt{fin}\rangle)$ is still not e-verified, we keep the status $\mathtt{fin}$. Let $s_0 < s_1 < ... < s_n = s+1$ be the set of stages at which $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ has been activated after the last time u when $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ had outcome ∞. Let $t_i = s_i - u$ and let $\{\tau_0, ..., \tau_{k-1}\} = [S_{f(\langle\sigma,\alpha\rangle)}][u] \setminus \{r(\langle\sigma,\alpha^\frown\mathtt{fin}\rangle)\}$. For each $j < k$, define $[S_{\tau_j}][s+1] = \Gamma_{T_\sigma}(t_0, ..., t_n, [S_{\tau_j}][s])$, where Γ is as in Lemma 2.3. Activate the requirements $\mathcal{T}_{\langle\sigma^\frown 0,\alpha^\frown\mathtt{fin}\rangle}$ and $\mathcal{T}_{\langle\sigma^\frown 1,\alpha^\frown\mathtt{fin}\rangle}$.
  - Suppose now that $r(\langle\sigma,\alpha\rangle)$ has been e-verified since the last time $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ was active. Let $o_{s+1}(\sigma) = \infty$. Define $r(\langle\sigma,\alpha^\frown\infty\rangle)[s+1] = f(\langle\sigma,\alpha\rangle)$. Let u be the last stage when $\mathcal{S}_{\langle\sigma,\alpha\rangle}$'s status was ∞ and $\{\tau_0, ..., \tau_k\} = [S_{f(\langle\sigma,\alpha\rangle)}][u]$. If τ_i is of the form $r(\langle\pi,\delta\rangle)[u]$, then let $r(\langle\pi,\delta\rangle)[s]$ be an extension of it in $[S_{f(\langle\sigma,\alpha\rangle)}][s]$. Activate requirements $\mathcal{T}_{\langle\sigma^\frown 0,\alpha^\frown\infty\rangle}$ and $\mathcal{T}_{\langle\sigma^\frown 1,\alpha^\frown\infty\rangle}$.

$\Diamond$

Suppose $\langle\sigma,\alpha\rangle \in \mathtt{TP}$ and $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ gets activated infinitely often and after some stage s_0 it is never canceled. If after some stage s_1, whenever $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ is activated, its outcome is $\mathtt{fin}$, then we will have that $r(\sigma) = r(\langle\sigma, o_{s_1}\restriction\sigma+\rangle)$ and $[r(\sigma)] \subseteq F_e$. So $[S_{r(\sigma)}] \subseteq \int([F_e])$, and

$[S_{f(\sigma)}] = X_\sigma \cap [S_{r(\sigma)}]$, where X_σ is a finite union of set isomorphic to Q_σ, and hence X_σ is perfect and thin if $[T_\sigma]$ is perfect and X_σ is finite otherwise.

Otherwise, if $\mathcal{S}_{\langle\sigma,\alpha\rangle}$ has ∞ outcome infinitely many times, then there requirements extending $\langle\sigma, \alpha^\frown\infty\rangle$ will have the chance to act infinitely often and will never be canceled by $\mathcal{S}_{\langle\sigma,\alpha\rangle}$. In this case we have that every sting of $S_{f(\sigma)}$ gets e-verified and hence $[S_{f(\sigma)}] \cap int([F_e]) = \emptyset$.

4.4. **Thinness requirements.** For each $\langle\sigma, \alpha\rangle \in TS$, of length $3e$ we have a thinness requirement: either $[(F_e)_{f(\sigma)}] = [S_{f(\sigma)}]$, or $[F_e] \cap [S_{f(\sigma)}] = \emptyset$. It does exactly the same as the module described in Section 2.2. As in the previous constructions, it will not build any set X_σ and we will have $[S_{f(\sigma)}] = [S_{r(\sigma)}]$. When it is initialized is stars on status $\mathtt{wai}$, and $r(\langle\sigma, \alpha^\frown\mathtt{wai}\rangle) = f(\langle\sigma, \alpha\rangle)$. If it ever acts, it changes its status to $\mathtt{sat}$ and defines $r(\langle\sigma, \alpha^\frown\mathtt{sat}\rangle)$ to be some string not in F_e and it is then satisfied for ever. Note that if $\mathcal{T}_{\langle\sigma,\alpha\rangle}$ is in status $\mathtt{sat}$ and some other requirement changes the value of $r(\langle\sigma, \alpha^\frown\mathtt{sat}\rangle)$ for a longer string, then $\mathcal{T}_{\langle\sigma,\alpha\rangle}$ reminds satisfied. So, as in the comments at the end of Section 2.2, so long as $\mathcal{T}_{\langle\sigma,\alpha\rangle}$ gets to act infinitely often after the last time it was initialized, it will be satisfied. The reason why all the requirements $\mathcal{T}_\sigma$ for σ of length $3e$ imply $\mathcal{T}_e$ is the same as in Section 3.2.

4.5. **Isolation requirements.** For each $\sigma \in T$, σ of length $3e + 1$ we have a isolation requirement: if $[T_\sigma]$ is isolated, $[S_{f(\sigma)}]$ is isolated . It does exactly the same as the module described in the previous construction, and so long as it gets to act infinitely often without being canceled, it will make sure that (Sfr2) is satisfied as in the previous construction.

4.6. **Verifications.** Let D be the set of $\sigma \in T$ such that $[T_\sigma]$ is non-empty and $[T_{\sigma^-}]$ is not isolated.

Lemma 4.2. *Consider $\sigma \in D$.*

(1) *There is a stage s_0, after which $\mathcal{R}_{\langle\sigma,TP\restriction\sigma\rangle}$ is activated infinitely often and never canceled again. We also have that $f(\sigma)$ exists and equals $f(\langle\sigma, TP\restriction\sigma\rangle[s_0]$.*
(2) *$\mathcal{R}_\sigma$ is satisfied.*
(3) *For σ not an end-node of D, $\mathcal{R}_{\langle\sigma,TP\restriction\sigma\rangle}$ acts only finitely often, and $r(\sigma)$ exists.*
(4) *$TP(\sigma)$ exists.*

Proof. The proof is by simultaneous induction on the length of σ. □

References

[1] Binns, S., *Small Π^0_1 classes,* to appear.
[2] Cenzer, D., Π^0_1 *classes in computability theory,* in *Handbook of Computability*, Elsevier, Amsterdam, 1999, 37-85.
[3] Cenzer, D., Downey, R., Jockusch C., and Shore, R. A., *Countable thin Π^0_1 classes,* Annals of Pure and Applied Logic, 59 (1993) 79-139,
[4] Cenzer, D., and Jockusch, C., Π^0_1*-Classes-Structure and applications,* in *Computability Theory and its Applications,* (ed. P. Cholak, S. Lempp, M. Lerman, and R. Shore) Contemporary Mathematics, Vol. 257, AMS Publications, Rhode Island, 2000, 39-60.
[5] Cenzer, D., and A. Nies, *Global properties of the lattice of* Π^0_1 *classes*, Proceedings American Mathematical Society 132 (2004), 239-249.
[6] Cenzer, D., and Remmel, J., Π^0_1 *classes in mathematics,* in *Handbook of Recursive Mathematics, Vol II* (ed. Y. Ershov, S. Goncharov, A. Nerode, and J. Remmel), Elsevier, Amsterdam, (1998), 623-822.

[7] Cholak, P., R. Coles, R. Downey, and E. Herrmann, *Automorphisms of the lattice of* Π^0_1 *classes: perfect thin classes and ANC degrees,* Trans. Amer. Math. Soc., Vol. 353 (2001), 4899-4924.

[8] Cholak, P, and R. Downey, *Invariance and noninvariance in the lattice of* Π^0_1 *classes,* Journal of the London Mathematical Society (2) Vol. 70 (2004), 735-749.

[9] Downey, R., *Abstract Dependence, Recursion Theory and the Lattice of Recursively Enumerable Filters* Thesis, Monash University, Clayton, Victoria, Australia, (1982).

[10] Downey, R., *Maximal theories,* Annals of Pure and Applied Logic, 33 (1987) 245-282.

[11] Downey, R., C. Jockusch, and M. Stob, *Array nonrecursive sets and multiple permitting arguments,* in *Recursion Theory Week* (Ambos-Spies, Muller, Sacks, eds.) Lecture Notes in Mathematics 1432, Springer-Verlag, Heidelberg, 1990, 141–174.

[12] Feiner, L., *Hierarchies of Boolean algebras*, Journal of Symbolic Logic, 35 (1970), 365-374.

[13] Jockusch, C. and Soare, R. I., Π^0_1 *classes and degrees of theories,* Trans. Amer. Math. Soc. 142 (1969) 229-237.

[14] Lachlan, A. H., Theorem XV 2.2 in Soare [22].

[15] Martin, D., *Classes of recursively enumerable sets and degrees of unsolvability,* Z. Math. Logik Grundlag. Math. 12 (1966) 295-310.

[16] Martin, D. and Pour-El, M., *Axiomatizable theories with few axiomatizable extensions,* J. Symbolic Logic 35 (1970) 205-209.

[17] Nies, A., F. Stephan and S. A. Terwijn, *Randomness, relativization, and Turing degrees*, Journal of Symbolic Logic, Vol. 70(2), (2005), 515-535.

[18] Remmel, J. B., *Recursively enumerable boolean algebras,* Ann. Math. Logic, vol. 14 (1978), 75–107.

[19] Remmel, J. B., *Graph coloring and recursively bounded* Π^0_1 *classes,* Ann. Pure and Appl. Logic, vol. 32 (1986), 185–194.

[20] Simpson, S., *Mass problems and randomness,* Bulletin of Symbolic Logic, Vol. 11 (2005), 1-27.

[21] Soare, R. I., *Automorphisms of the lattice of recursively enumerable sets I: maximal sets,* Annals of Math. (2), 100 (1974) 80-120.

[22] Soare, R. I., *Recursively enumerable sets and degrees,* Springer-Verlag New York (1987).

E-mail address: `Rod.Downey@mcs.vuw.ac.nz`
URL: http://www.mcs.vuw.ac.nz/∼downey

E-mail address: `antonio@mcs.vuw.ac.nz`
URL: www.math.uchicago.edu/∼antonio

School of Mathematics, Statistics and Computer Science, Victoria University, P.O. Box 600, Wellington, New Zealand