# CONSTRUCTING SEQUENCES ONE STEP AT A TIME

HENRY TOWSNER

ABSTRACT. We propose a new method for constructing Turing ideals satisfying principles of reverse mathematics below the Chain-Antichain Principle (**CAC**). Using this method, we are able to prove several new separations in the presence of Weak König's Lemma (**WKL**), including showing that **CAC** + **WKL** does not imply the thin set theorem for pairs, and that the principle "the product of well-quasi-orders is a well-quasi-order" is strictly between **CAC** and the Ascending/Descending Sequences principle, even in the presence of **WKL**.

## 1. INTRODUCTION

**Definition 1.1.** A *Turing ideal* is a collection $\mathcal{I}$ of sets such that whenever $X \in \mathcal{I}$ and the set $Y$ is computable from $X$, also $Y \in \mathcal{I}$.

The principles we discuss here are usually formulated in the context of reverse mathematics, but since that formulation will not be needed here, we state them in terms of Turing ideals. (Those familiar with reverse mathematics [9] will recognize that our main concern is constructing $\omega$-models witnessing various separations.) We are interested in Turing ideals which exhibit certain closure properties: ideals $\mathcal{I}$ so that whenever $X \in \mathcal{I}$ encodes an *instance* of problem a certain kind, $\mathcal{I}$ also contains some $Y$ which is a *solution* to that instance.

An important example is:

**Definition 1.2.** A Turing ideal $\mathcal{I}$ satisfies **WKL** ("Weak König's Lemma") if whenever $T \in \mathcal{I}$ encodes an infinite tree of $\{0,1\}$ sequences, there is an infinite $\{0,1\}$ sequence $\Lambda \in \mathcal{I}$ so that for every $n$, $\Lambda \upharpoonright n \in T$.

**Definition 1.3.** We say that a principle **P** *implies* **Q** if any Turing ideal satisfying **P** also satisfies **Q**.

All our other principles concern weakenings or variants of Ramsey's Theorem for pairs. Recall that Ramsey's Theorem for pairs says that whenever $c : [\mathbb{N}]^2 \to \{0,1\}$ is a coloring of pairs, there is an infinite homogeneous set: an infinite set $S \subseteq \mathbb{N}$ and an $i$ so that whenever $a, b \in S$, $c(a,b) = i$

---

*Date*: January 17, 2017.

Partially supported by NSF grant DMS-1600263.

The work was begun while the author was visiting the Institute for Mathematical Sciences, National University of Singapore in 2016. The visit was supported by the Institute.

Most of the weakenings we are interested in concern partial or total orders. An ordering $\prec$ can be associated with a coloring by setting $c(a,b) = 1$ iff $a \prec b$ (where we assume $a, b$ are ordered $a < b$ in the usual ordering on the natural numbers).

**Definition 1.4.** A Turing ideal $\mathcal{I}$ satisfies **CAC** ("Chain-Antichain") if whenever $\preceq$ is an $\mathcal{I}$-computable partial ordering, there is an infinite sequence $\Lambda$ in $\mathcal{I}$ which is either $\prec$-increasing, $\prec$-decreasing, or an antichain in $\prec$.

This is equivalent to restricting Ramsey's Theorem for pairs to the special case where one of the colors is transitive [4].

**Definition 1.5.** If $c : [\mathbb{N}]^2 \to \mathbb{N}$ is a coloring, we say a color $i$ is *transitive* if whenever $a_0 < a_1 < a_2$ with $c(a_0, a_1) = c(a_1, a_2) = i$, also $c(a_0, a_2) = i$.

A natural further restriction is to ask that $\preceq$ be a linear ordering.

**Definition 1.6.** A Turing ideal $\mathcal{I}$ satisfies **ADS** ("Ascending/Descending Sequences") if whenever $\prec$ is an $\mathcal{I}$-computable linear ordering, there is an infinite sequence $\Lambda$ in $\mathcal{I}$ which is either $\prec$-increasing or $\prec$-decreasing.

This is slightly stronger than requiring that both colors be transitive, but is equivalent at the level of Turing ideals.

**Definition 1.7.** A Turing ideal $\mathcal{I}$ satisfies $\mathbf{trRT^2_k}$ ("transitive Ramsey's Theorem for pairs with $k$ colors") if whenever $c : [\mathbb{N}]^2 \to [1, k]$ is a coloring where all colors are transitive, there is an infinite set $S$ and an $i \in [1, k]$ so that whenever $a, b \in S$, $c(a, b) = i$.

The basic relationships between **CAC**, **ADS**, and $\mathbf{trRT^2_k}$ are set out in [4].

**Lemma 1.8** ([4])**.** *A Turing ideal satisfies* **ADS** *iff it satisfies* $\mathbf{trRT^2_2}$.

Furthermore, **CAC** implies $\mathbf{trRT^2_k}$ for any $k$, and $\mathbf{trRT^2_{k+1}}$ implies $\mathbf{trRT^2_k}$.

Showing that these implications do not reverse is more difficult. Lerman, Solomon, and Towsner constructed a Turing ideal satisfying **ADS** but not **CAC** [5], and Patey showed that a similar method can construct a Turing ideal satisfying $\mathbf{trRT^2_k}$ but not **CAC** [8]. (More precisely, Patey studies a principle shown to be very similar in [6].) It is not known whether $\mathbf{trRT^2_k}$ implies $\mathbf{trRT^2_{k+1}}$.

Dzhafarov, Goh, and Shore asked whether these separations remain in the presence of **WKL**. As we will discuss in detail below, satisfying **WKL** appears to conflict with the method used in [5], and a new approach to the separation is required. Using this approach, we will show:

**Theorem 1.9.** *There is a Turing ideal satisfying* $\mathbf{trRT^2_k}$ *for all $k$ and* **WKL** *but not* **CAC**.

While considering this question, one naturally considers what else might be a consequence of **ADS** together with **WKL**. In particular, one asks whether these principles might imply other consequences of Ramsey's Theorem for pairs which do not follow from **CAC**. For example:

**Definition 1.10.** A Turing ideal $\mathcal{I}$ satisfies $\mathbf{TS(2)}$ ("Thin Sets for Pairs") if whenever $c : [\mathbb{N}]^2 \to \mathbb{N}$ is $\mathcal{I}$-computable function, there is an infinite set $S$ in $\mathcal{I}$ and a color $i$ so that there is no $x, y \in S$ with $c(x, y) = i$.

This thin set principle was introduced in [3] and further studied in [1, 7, 10].

Using a similar method, we are able to show:

**Theorem 1.11.** *There is a Turing ideal satisfying* **CAC** *and* **WKL** *but not* $\mathbf{TS(2)}$*.*

Hirschfeldt and Shore ask [4] whether the $\mathbf{trRT^2_k}$ hierarchy is strict.

**Question 1.12.** *Does* $\mathbf{trRT^2_k}$ *imply* $\mathbf{trRT^2_{k+1}}$*?*

Normally adding more colors does not change the difficulty of satisfying a Ramsey theoretic principle: one "merges" two of the colors into a single color and then applies the Ramsey theoretic argument repeatedly. But this fails with $\mathbf{trRT^2_k}$ because the merged color may not be transitive.

Asking how we should strengthen the statement to allow such a merger of colors leads us to define:

**Definition 1.13.** A Turing ideal $\mathcal{I}$ satisfies **ProdWQO** ("Products of WQOs are WQO") if whenever $c : [\mathbb{N}]^2 \to \{0, 1, 2\}$ and the colors 1 and 2 are transitive, there is an infinite set $S$ and an $i \in \{1, 2\}$ so that whenever $a, b \in S$, $c(a, b) \neq i$.

(The name will be justified below.) That is, we have a coloring with two transitive colors and one color which need not be transitive where we can always *omit* one of the transitive colors.

**Lemma 1.14** ([2])**. CAC** *implies* **ProdWQO***.*

Frittaion, Marcone, and Shafer pointed out that **ProdWQO** implies **ADS**.

**Lemma 1.15. ProdWQO** *implies* $\mathbf{trRT^2_k}$ *for any $k$, and so also* **ADS***.*

*Proof.* Let $c : [\mathbb{N}]^2 \to [1, k]$ be a transitive coloring. For any pair $i \neq j$ in $[1, k]$, define the coloring $c_{i,j} : [\mathbb{N}]^2 \to [0, 1, 2]$ given by

$$c_{i,j}(a, b) = \begin{cases} 1 & \text{if } c(a, b) = i \\ 2 & \text{if } c(a, b) = j \\ 0 & \text{otherwise} \end{cases}$$

By **ProdWQO** applied to $c_{1,2}$, we have an infinite set $S$ omitting either color 1 or color 2; without loss of generality, we assume $S$ omits 1. Applying **ProdWQO** to $c_{2,3}$ (more precisely, let $\pi : \mathbb{N} \to S$ be the unique injective, order-preserving map, define $c'_{2,3}(i,j) = c_{2,3}(\pi(i),\pi(j))$, and apply **ProdWQO** to $c'_{2,3}$) restricted to the set $S$, we omit a second color. We iterate this until only one color is remaining, at which point the set must be homogeneous. $\square$

Although we phrase it here in terms of transitive colorings, **ProdWQO** is more naturally seen as the statement that a product of well-quasi-orders is also well-quasi-ordered. Recall that a partial ordering $\preceq$ is well-quasi-ordered if whenever $\langle a_1, a_2, \ldots\rangle$ is an infinite sequence, there exist $i < j$ so that $a_i \preceq a_j$. An infinite sequence $\langle a_1, a_2, \ldots\rangle$ is *bad* if it witnesses the failure to be a well-quasi-order: whenever $i < j$, $a_i \not\preceq a_j$.

The product $\preceq = \preceq_1 \times \preceq_2$ of two quasi-orderings is given by $a \preceq b$ iff both $a \preceq_1 b$ and $a \preceq_2 b$. To say that the product of two well-quasi-orders is also well-quasi-ordered is the same as saying that whenever we have a product $\preceq = \preceq_1 \times \preceq_2$ and an infinite bad sequence in $\preceq$ then we must have an infinite bad sequence in either $\preceq_1$ or in $\preceq_2$. If we define a coloring

$$c(a,b) = \begin{cases} 1 & \text{if } a \preceq_1 b \\ 2 & \text{if } a \preceq_2 b \\ 0 & \text{otherwise} \end{cases}$$

then this is well-defined on an infinite bad sequence (because we cannot have both $a \preceq_1 b$ and $a \preceq_2 b$). The colors 1 and 2 are transitive while 0 need not be. Finding a bad sequence in $\preceq_i$ exactly means finding an infinite sequence avoiding $i$, which is precisely what our formulation of **ProdWQO** says.

Our remaining results show that **ProdWQO** is properly intermediate between **ADS** and **CAC**.

**Theorem 1.16.**
- *There is a Turing ideal satisfying* $\mathbf{trRT^2_k}$ *for all* $k$ *and* **WKL** *but not* **ProdWQO**.
- *There is a Turing ideal satisfying* **ProdWQO** *and* **WKL** *but not* **CAC**.

Of course, either of these results implies Theorem 1.9.

Finally, we note that all these principles have a stable version.

**Definition 1.17.** A coloring of pairs $c : [\mathbb{N}]^2 \to \mathbb{N}$ is *stable* if for every $a$ there are $i$ and $j$ so that whenever $j \leqslant b$, $c(a,b) = i$.

**SADS** (respectively **SCAC**, $\mathbf{STS(2)}$, **SProdWQO**, $\mathbf{StrRT^2_k}$) is the principle **ADS** (respectively **CAC**, $\mathbf{TS(2)}$, **ProdWQO**, $\mathbf{trRT^2_k}$) restricted to stable instances.

In fact, all our results also apply to the stable versions of these principles; that is, when we show that we fail to satisfy a principle, we always fail to satisfy a stable instance.

The author is grateful to Frittaion, Marcone, and Shafer for pointing out that **ProdWQO** is between **ADS** and **CAC** and raising the question of where it fits. Some of the ideas leading to the work here were developed

in discussions with Kuyper, Lempp, Miller, and Soskova. Finally, Patey provided feedback and suggestions on a long strong of initial attempts at this work, including pointing the author towards the crucial obstacles and suggesting several ways that the results in this paper could be strengthened.

## 2. Separating $\mathbf{STS(2)}$

In this section we construct a computable instance $c$ of $\mathbf{STS}(2)$ and then construct a Turing ideal $\mathcal{I}$ which has no solution to $c$, but does satisfy both **CAC** and **WKL**.

Since this is the prototype for our other arguments, we take a moment to outline the structure. The ideal $\mathcal{I}$ will be defined by recursively building a sequence $I_1, I_2, \dots$ of sets and taking $\mathcal{I}$ to be those things computable from $\oplus_{i\leqslant n} I_i$ for some $n$. Given $X = \oplus_{i\leqslant n} I_i$ for some $n$, we will define the notion of a *requirement* (computable) in $X$, and the notion of when a particular instance $c$ of $\mathbf{STS}(2)$ *satisfies* a given requirement in an oracle $X$. We will then prove:

(1) if $c$ satisfies all requirements in $X$ then there is no $X$-computable solution to $c$ (Lemma 2.8),
(2) if $c$ satisfies all requirements in $X$ and $\preceq$ is an $X$-computable partial ordering then there is an infinite chain or antichain $\Lambda$ so that $c$ satisfies all requirements in $X \oplus \Lambda$ (Lemma 2.13),
(3) if $c$ satisfies all requirements in $X$ and $U$ is an infinite $X$-computable $\{0,1\}$-branching tree then there is an infinite branch $\Lambda$ so that $c$ satisfies all requirements in $X \oplus \Lambda$ (Lemma 2.14), and
(4) there exists a computable stable $c$ satisfying all requirements in $\varnothing$ (Lemma 2.16).

These four pieces give the desired result:

**Theorem 2.1.** *There is a computable stable $c : [\mathbb{N}]^2 \to \mathbb{N}$ and a Turing ideal $\mathcal{I}$ so that:*

- *if $I \in \mathcal{I}$ is infinite then $c \upharpoonright [I]^2 = \mathbb{N}$,*
- *$\mathcal{I}$ satisfies* **CAC***, and*
- *$\mathcal{I}$ satisfies* **WKL***.*

*Proof.* We take the $c$ given by Lemma 2.16 and then use Lemma 2.13 and Lemma 2.14 to recursively define the sets $I_i$ so that $c$ satisfies all requirements in $\oplus_{i\leqslant n} I_i$, so that if $\preceq$ is an $\oplus_{i\leqslant n} I_i$-computable partial ordering the there is some $k$ so that $I_k$ is an infinite chain or antichain, and so that if $U$ is an infinite $\oplus_{i\leqslant n} I_i$-computable $\{0,1\}$-branching tree then there is some $k$ so that $I_k$ is an infinite branch of $U$. Then the Turing ideal consisting of all sets computable from $\oplus_{i\leqslant n} I_i$ for some $n$ will have the desired properties. $\square$

### 2.1. Requirements.

**Definition 2.2.** Let $c : [\mathbb{N}]^2 \to \mathbb{N}$ be stable. For each $i$, $A^*_i(c)$ consists of those $n$ so that, for cofinitely many $m$, $c(n, m) = i$.

Clearly the $A_i^*(c)$ are disjoint; stability implies that they form a partition of $\mathbb{N}$.

**Definition 2.3.** A *simple block statement in $X$* is a set computable from an oracle $X$ of the form $K^X(b,\vec{a})$ (with the groups of variables distinguished).

The parameters are intended as follows:

- $b$ is an auxiliary datum,
- $\vec{a}$ is a set of witnesses which might be in $A_i^*(c)$ for some $i$.

**Definition 2.4.** A *requirement* $R = (T, \{K_\sigma\}_{\sigma\in T}, \{d_\sigma\}_{\sigma\in T})$ is a finite, finitely branching tree $T$, for each $\sigma \in T$ a simple block statement $K_\sigma$ and a function $d_\sigma : \mathrm{dom}(\sigma) \to \mathbb{N}$, and so that $K_{\langle\rangle}$ is always true.

For any $\sigma \in T$, any $c : [\mathbb{N}]^2 \to \mathbb{N}$, and any oracle $X$, the *positive requirement component at $\sigma$* is the formula $\Delta^X_{R;\sigma}(c, b_0, \ldots, b_{|\sigma|-1}, \vec{a}_0, \ldots, \vec{a}_{|\sigma|-1})$ which holds if, for each $i < |\sigma|$, $K^X_{\sigma\upharpoonright(i+1)}((b_0, \ldots, b_i), \vec{a}_i)$ holds.

If $\sigma \in T$ is a leaf, $\Theta^X_{R;\sigma}(c)$ is the formula which holds if there exist $b_0, \ldots, b_{|\sigma|-1}, \vec{a}_0, \ldots, \vec{a}_{|\sigma|-1}$ so that:

- $\vec{a}_i \in A^*_{d_\sigma(i)}(c)$,
- $\Delta^X_{R;\sigma}(c, b_0, \ldots, b_{|\sigma|-1}, \vec{a}_0, \ldots, \vec{a}_{|\sigma|-1})$ holds.

If $\sigma \in T$ is not a leaf, $\Theta^X_{R;\sigma}(c)$ is the formula which holds if there exist $b_0, \ldots, b_{|\sigma|-1}, \vec{a}_0, \ldots, \vec{a}_{|\sigma|-1}$ and a $t$ so that:

- $\vec{a}_i \in A^*_{d_\sigma(i)}(c)$,
- $\Delta^X_{R;\sigma}(c, b_0, \ldots, b_{|\sigma|-1}, \vec{a}_0, \ldots, \vec{a}_{|\sigma|-1})$,
- there do not exist $b, \vec{a}$, and $\tau$ an immediate extension of $\sigma$ in $T$ so that $t < \vec{a}$ and $\Delta^X_{R;\tau}(c, b_0, \ldots, b_{|\sigma|-1}, b, \vec{a}_0, \ldots, \vec{a}_{|\sigma|-1}, \vec{a})$.

We say *$c$ satisfies a requirement $R = (T, \{K_\sigma\}_{\sigma\in T}, \{d_\sigma\}_{\sigma\in T})$ in $X$* if there is some $\sigma \in T$ so that $\Theta^X_{R;\sigma}(c)$ holds.

We will sometimes wish to work with requirements satisfying certain restrictions.

**Definition 2.5.** A requirement $R = (T, \{K_\sigma\}_{\sigma\in T}, \{d_\sigma\}_{\sigma\in T})$ has *range $I$* if for every $\sigma \in T$, $\mathrm{rng}(d_\sigma) \subseteq I$.

A requirement $R = (T, \{K_\sigma\}_{\sigma\in T}, \{d_\sigma\}_{\sigma\in T})$ is *transitive in color $i$* if whenever $\tau \subsetneq \sigma$, $j < |\tau|$, $d_\tau(j) = i$, and $d_\sigma(|\tau|) = i$, then $d_\sigma(j) = i$.

While we mostly find it natural to work with trees of requirement, we note that it does suffice to consider linear ones.

**Definition 2.6.** A requirement is *linear* if $\sigma \in T$ implies $\sigma$ has the form $\langle 0, 0, \ldots, 0\rangle$.

**Lemma 2.7.** *Suppose $c$ satisfies every linear requirement in $X$ with range $I$ which is transitive in every color in $J \subseteq I$ where $0 \in I\backslash J$. Then $c$ satisfies every requirement in $X$ with range $I$ which is transitive in every color in $J$.*

*Proof.* Let $R = (T, \{K_\sigma\}_{\sigma\in T}, \{d_\sigma\}_{\sigma\in T})$ be a requirement with range $I$ which is transitive in every color in $J \subseteq I$. We define a linear requirement whose satisfaction ensures that we have satisfied $R$.

Let $n = |T|$ and fix a function $\pi : T \to [0, n)$ so that $\sigma \sqsubseteq \tau$ implies $\pi(\sigma) \leqslant \pi(\tau)$. We let $T'$ consist of sequence of the form $\langle 0, \ldots, 0\rangle$ with length $< n$ and we associate the sequence in $T'$ of length $i$ with the natural number $i$.

When $j < |\sigma|$, we set $d_{\pi(\sigma)}(\pi(\sigma \upharpoonright j)) = d_\sigma(j)$, and $d_{\pi(\sigma)}(j) = 0$ otherwise. This ensures that $T'$ will have the same range and satisfy the same transitivity requirements, as needed.

The auxiliary data will have the form $(r_i, b_i)$ where $r_i$ is either an element of $T$ or 0. $(K')^X_i(((r_0, b_0), \ldots, (r_{i-1}, b_{i-1})), \vec{a}_0, \ldots, \vec{a}_{i-1})$ holds if, letting $i_1, \ldots, i_k < i$ be those values such that $r_{i_j} \neq 0$:

- $k \geqslant 0$ (i.e. there is at least one such $i$ with $r_{i_j} \neq 0$),
- $r_{i_j}$ is a sequence with $|r_{i_j}| = j$,
- $r_{i_0} \subsetneq r_{i_1} \subsetneq \cdots \subsetneq r_{i_k}$,
- if $0 < j < k$ then $i_{j+1} = \pi(r_{i_j})$,
- $K^X_{r_{i_k}}((b_{i_0}, \ldots, b_{i_k}), \vec{a}_{i_0}, \ldots, \vec{a}_{i_k})$,
- $\pi(r_{i_k}) \geqslant i$.

Suppose $\Theta^X_{R';i}(c)$ holds for some $i$. Let $\sigma = \pi^{-1}(i)$, and let $(r_0, b_0)$, …, $(r_{i-1}, b_{i-1})$, $\vec{a}_0$, …, $\vec{a}_{i-1}$ be the witnessing data. Let $i_1, \ldots, i_k < i$ be the witnesses; note that if $\pi(r_{i_k}) > i$ then we would also satisfy $\Theta^X_{R';i+1}(c)$, so we may assume either $i = 0$ (so $\sigma = \langle\rangle$) or $\sigma = r_{i_k}$. So for any $\tau \sqsubseteq \sigma$, we have $\tau = r_{i_{|\tau|}}$, so $K^X_\tau((b_{i_0}, \ldots, b_{i_{|\tau|}}), \vec{a}_{i_0}, \ldots, \vec{a}_{i_{|\tau|}})$.

On the other hand, if there were some immediate extension $\sigma$ of $r_{i_k}$, a $b$, and a $\vec{a}$ so that $\Delta^X_{R;\sigma}(c, b_{i_0}, \ldots, b_{i_k}, b, \vec{a}_{i_0}, \ldots, \vec{a}_{i_k}, \vec{a})$ holds then $(\sigma, b), \vec{a}$ would witness $(K')^X_{i+1}$. So we have $\Theta^X_{R;r_{i_k}}(c)$. □

**Lemma 2.8.** *Suppose $c$ satisfies every requirement in $X$ with range $I$ which is transitive in every color in $J \subseteq I$ where $0 \in I\backslash J$. Then whenever $B$ is an $X$-computable (or even $X$-computably enumerable) infinite set, $c \upharpoonright [B]^2 \supseteq I$.*

*Proof.* For each $e$ and each $i \in I$, we show that if $W_e$ is infinite then there is an $x \in W_e \cap A^*_i(c)$; then since $W_e$ is infinite, there must be a big enough $x \in W_e$ with $c(x, y) = i$.

We take $T$ to contain a single branch of length 1, $\langle 0\rangle$. We take $K^X_{\langle 0\rangle}(b, x)$ to hold if $x \in W^X_{e,b}$. We set $d_{\langle 0\rangle}(0) = i$. If $\Theta_{R;\langle\rangle}(c)$ holds then there must be some $t$ so that there do not exist $b$ and $x > t$ so that $x \in W^X_{e,b}$; but this implies that $W_e$ is finite. Otherwise $\Theta_{R;\langle 0\rangle}(c)$ holds, in which case we find $b_0, x_0$ so that $x_0 \in W^X_{e,b_0}$ and $x_0 \in A^*_i(c)$ as needed. □

Before going on, we attempt to motivate our definition of a requirement. Our discussion will be most meaningful to someone already familiar with the construction in [5]. For purposes of this discussion, we consider a separation easier than any of the others considered in this paper: separating **ADS** from $\mathbf{D}^2_2$; the latter is $\mathbf{STS}(2)$ restricted to the colors $\{0,1\}$, where a solution must omit one of these colors (and therefore be homogeneous in the other color).

We imagine that we are simultaneously constructing our instance $c$ of $\mathbf{D}^2_2$ and our solution to some instance $\prec$ of **ADS**, and we wish to make a single step of our construction, which means arranging progress towards either a $\prec$-increasing sequence $\Lambda^+$ so that $\Phi^{\Lambda^+}_{e_0}$ fails to compute a solution to $c$ or a $\prec$-decreasing sequence $\Lambda^-$ so that $\Phi^{\Lambda^-}_{e_1}$ fails to compute a solution to $c$. The key idea of [5] was to look for both a $\prec$-increasing sequence $p$ with endpoint $p^+$ and a $\prec$-decreasing sequence $q$ with endpoint $q^+$ so that:

- $p^+ \preceq q^+$,
- there are two fresh elements $a_0, a_1$ so that $\Phi^p_{e_0}$ converges and equals 1 on both $a_0$ and $a_1$,
- there are two fresh elements $b_0, b_1$ so that $\Phi^q_{e_1}$ converges and equals 1 on both $b_0$ and $b_1$.

If this happens, we could restrain $c$ so that we will have $a_0, b_0 \in A^*_0(c)$ and $a_1, b_1 \in A^*_1(c)$. Then, since $p^+ \preceq q^+$, *either* there are infinitely many $x$ with $p^+ \prec x$ (and therefore $p$ is a reasonable beginning of an increasing sequence), or there are infinitely many $x$ with $x \prec q^+$ (and therefore $q$ is a reasonable beginning of a decreasing sequence). Crucially, if we fail to find such a pair $p, q$, then one can arrange for either $\Phi^{\Lambda^+}_{e_0}$ or $\Phi^{\Lambda^-}_{e_1}$ to be finite.

The difficult point is that one needs to ensure $b_0 \neq a_1$ and $a_0 \neq b_1$ so that we can place both of the needed restraints separately.

This is the source of the conflict when one attempts to strengthen the separation by including solutions to **WKL**. One ends up working not with a single attempt at building $\Lambda^+$ and $\Lambda^-$, but with a finitely branching tree of attempts. The problem is that even if one finds such pairs $p, q$ in each branch, there may be incompatibilities across different branches — $a_0$ in one branch may be $b_1$ in another.

What one would prefer is to construct our witnesses in stages. First we would look for a pair $p_0, q_0$ with $p^+_0 \prec q^+_0$ and only the witnesses $a_0, b_0$. Then we could look for extensions $p_0 \sqsubseteq p_1$ and $q_0 \sqsubseteq q_1$ with $p^+_1 \preceq q^+_1$, and demand that the witnesses $a_1, b_1$ be above some threshold based on the first stage (in particular, larger than $\max\{a_0, b_0\}$). Such a construction would be compatible with a finitely branching tree: we could wait for the pairs $p_0, q_0$ to appear in every branch. The witnesses $a_0, b_0$ taken over all branches would form a "block" which is all restrained in the same way (say, all put into $A^*_0(c)$). Only then would we look for the extensions $p_1, q_1$ in all branches, requiring that the witnesses $a_1, b_1$ all be larger than any element of the 0 block.

The difficulty is that we need the following property: suppose we find our witnesses $p_0, q_0$, but then are unable to extend to $p_1, q_1$. Then this must be a situation in which we can succeed (presumably by forcing one of $\Phi^{\Lambda^+}_{e_0}, \Phi^{\Lambda^-}_{e_1}$

to be finite), even if a different choice of $p_0, q_0$ could have been extended to a $p_1, q_1$.

Let us state this more explicitly, since it is the driving force behind our definition above. When we wish to satisfy some requirement, we will proceed in stages in which we look for auxiliary data (like $p_0, q_0$) and witnesses (like $a_0, b_0$). When we find the data and witnesses, we may "restrain" the witnesses (by placing them in some $A_i^*(c)$) and then begin looking for the next stage of the construction. However:

- during each stage, all witnesses found at a given earlier stage must be restrained the same way, and
- at each stage, failing to find the data and witnesses to the next stage must be sufficient to ensure our requirement.

This is essentially what our definition of satisfaction of a requirement says.

In fact, the two-stage construction we alluded to two paragraphs ago fails: having found the witnesses $p_0, q_0$, failing to find $p_1, q_1$ is not helpful. It could be that, say, $p_0^+$ will actually turn out to be quite large in $\prec$, and no further elements will appear above $p_0^+$, making the extension $p_1$ impossible to find, and also meaning that our inability to find it gives us no information about how to restrain $\Lambda^+$ to make $\Phi_{e_0}^{\Lambda^+}$ finite.

In Figure 1 we lay out a multi-stage process which is substantially more complicated (the version there involves as many as six consecutive steps) For example, the next stage after finding $p_0, q_0$ is to look for *either* a pair $p_1, q_0'$ with $p_0 \sqsubseteq p_1$, $p_1^+ \preceq (q_0')^+$, $p_1$ finds a witness $a_1$, and $q_0'$ finds a new witness $b_0'$, *or* a pair $p_0', q_1$ with $q_0 \sqsubseteq q_1$, $(p_0')^+ \preceq q_1^+$, $q_1$ finds a witness $b_1$, and $p_0'$ finds a new witness $a_0'$.

2.2. **Solving ADS.** As a warm up to dealing with **CAC** (and a preview of Lemma 3.5), we first show that we can solve instances of **ADS** while preserving requirements.

As in [5], it is convenient to restrict to a certain kind of linear ordering.

**Definition 2.9.** A linear ordering $(\mathbb{N}, \prec)$ is *stable-ish* if there is a non-empty initial segment $V$ so that $V$ has no maximum under $\prec$ and $\mathbb{N} \setminus V$ has no minimum under $\prec$.

**Lemma 2.10** ([5])**.** *If $(\mathbb{N}, \prec)$ is not stable-ish then there is an infinite monotone $\prec$-sequence computable from $\prec$.*

Note that there is no requirement that the set $V$ be computable from $\prec$.

**Lemma 2.11.** *Suppose $c$ satisfies every requirement in $X$ and $\prec$ is a stable-ish $X$-computable linear ordering. Then there is a monotone sequence $\Lambda$ so that $c$ satisfies every requirement in $X \oplus \Lambda$.*

*Proof.* Let $V$ witness that $\prec$ is stable-ish. When $p$ is a monotone sequence, we write $p^+$ for the final element of $p$.

We will force with *conditions*, which are pairs $(p, q)$ where $p$ is a $\prec$-increasing sequence in $\prec$, $q$ is a $\prec$-decreasing sequence in $\prec$, $p^+ \in V$, and

$q^+ \notin V$. (This of course implies that $p^+ \prec q^+$. Note that being a condition is generally not $X$-computable, since $V$ need not be $X$-computable.) We say a condition $(p', q')$ *extends* $(p, q)$ if $p \sqsubseteq p'$ and $q \sqsubseteq q'$. We say $(p, q)$ *forces $R$ on the increasing side* if whenever $\Lambda$ is an infinite, $\prec$-increasing sequence with $p \sqsubseteq \Lambda$ and $\Lambda \subseteq V$, $c$ satisfies $R$ in $X \oplus \Lambda$. Similarly, we say $(p, q)$ *forces $R$ on the decreasing side* if whenever $\Lambda$ is an infinite, $\prec$-decreasing sequence with $q \sqsubseteq \Lambda$ and $\Lambda \subseteq V$, $c$ satisfies $R$ in $X \oplus \Lambda$.

It suffices to show:

> $(*)$ Suppose $R^+$ and $R^-$ are requirements and $(p, q)$ is a condition. Then there is a condition $(p', q')$ extending $(p, q)$ which either forces $R^+$ on the increasing side or $R^-$ on the decreasing side.

For suppose we have shown this. Then we fix a list of requirements $R_i^+, R_i^-$ so that for any pair of requirements $R^+, R^-$, there is an $i$ with $R_i^+ = R^+, R_i^- = R^-$. We construct a sequence $(\langle\rangle, \langle\rangle) = (p_0, q_0), (p_1, q_1), \dots$ with $(p_{i+1}, q_{i+1})$ extends $(p_i, q_i)$, $(p_{2i+1}, q_{2i+1})$ either forces $R_i^+$ on the increasing side or $R_i^-$ on the decreasing side, $p_{2i}$ has length $\geqslant i$, and $q_{2i}$ has length $\geqslant i$. Let $\Lambda^+ = \bigcup p_i$ and $\Lambda^- = \bigcup q_i$. If $c$ does not satisfy every requirement in $X \oplus \Lambda^+$ then there is some $R^+$ which it fails to satisfy, and therefore for each $R^-$ there was an $i$ with $R_i^+ = R^+, R_i^- = R^-$, and therefore since $(p_{2i+1}, q_{2i+1})$ must not have forced $R^+$ on the increasing side, $(p_{2i+1}, q_{2i+1})$ forced $R^-$ on the decreasing side, and therefore $\Lambda^-$ satisfies every requirement in $X \oplus \Lambda^-$.

We now show $(*)$. Let a condition $(p, q)$ and requirements $R^+, R^-$ be given. Let $R^+ = (T^+, \{L_\sigma\}_{\sigma \in T^+}, \{d_\sigma^+\}_{\sigma \in T^+})$ and $R^- = (T^-, \{M_\tau\}_{\tau \in T^-}, \{d_\tau^-\}_{\tau \in T^-})$ be given. We will describe a requirement $R = (T, \{K_\upsilon\}_{\upsilon \in T}, \{d_\upsilon\}_{\upsilon \in T})$.

For bookkeeping reasons, it is convenient to assume that for any $\sigma \in T^+$, $d_\sigma^+(|\sigma| - 1) = 0$; this is easily arranged: if $\sigma \in T^+$ violates this, modify $R^+$ as follows: insert a child $\sigma^\frown\langle 0\rangle$ so $L_{\sigma^\frown\langle 0\rangle}$ always holds, and wait for this dummy node to set $d^+_{\sigma^\frown\langle 0\rangle} = d_\sigma^+ \cup \{(|\sigma|, 0)\}$, then take all children $\sigma^\frown\gamma$ and move them to $\sigma^\frown\langle 0\rangle^\frown\gamma$. Symmetrically, we make the same assumption for $R^-$.

A *split pair* is a pair $(p', q')$ so that $p \sqsubseteq p'$, $q \sqsubseteq q'$, and $(p')^+ = (q')^+$. Note that a split pair need not be a condition, but being a split pair *is* $X$-computable. Crucially, when $(p', q')$ is a split pair, one of $(p, q')$ and $(p', q)$ must be a condition (depending on whether the common endpoint belongs to $V$).

First, we illustrate our construction in the simplest case that illustrates the general argument, where $T^+ = T^- = \{\langle\rangle, \langle 0\rangle, \langle 0, 0\rangle\}$. In Figure 1 we have indicated the requirement $R$ in this case, along with most of the auxiliary data associated with the node. The diagram requires some explanation. Each rectangle indicates a node of $T$, with the lines flowing from each parent node to its children.

Within a node, the paired arrows indicate split pairs. The length of the arrow indicates the length of the node from $R^+$ or $R^-$ which that

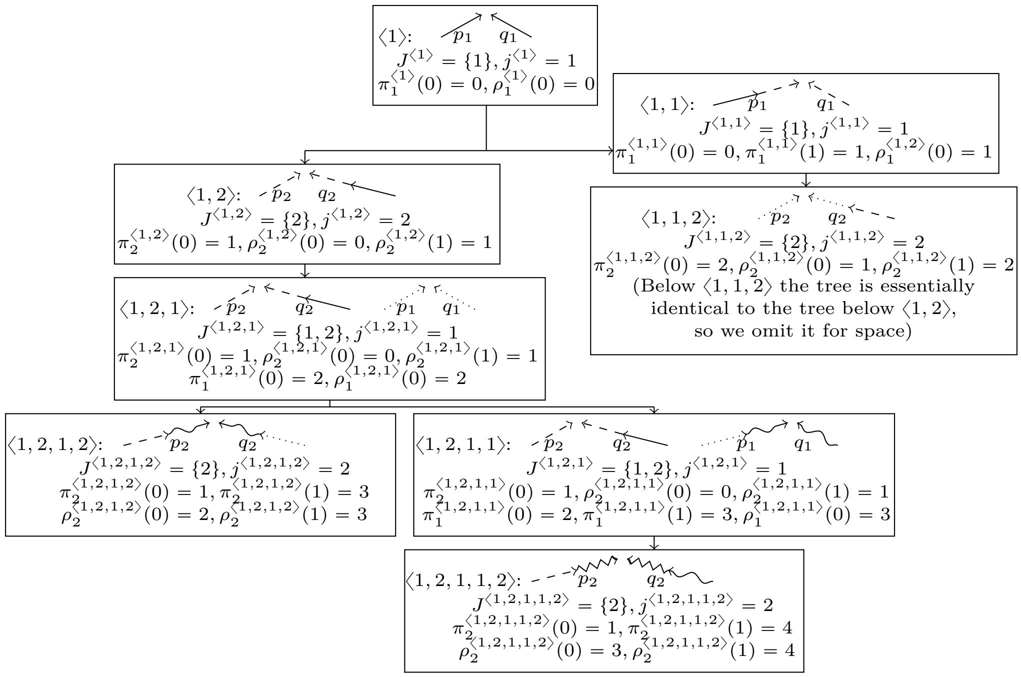


FIGURE 1.

element of the split pair witnesses; for instance, in node $\langle 1,2,1,2\rangle$, the arrow $p_2$ has length 2, indicating that we have found witnesses so that $\Delta^{X\oplus p_1}_{R^+;\langle 0,0\rangle}(c,(e_0,e_1),\vec{a}'_0,\vec{a}'_1)$ holds. For bookkeeping reasons, the subscript in $p_2$ indicates the length of the paired descending sequence $q_2$. The functions $\pi$ indicate which stage of our construction the witnesses were found at; for instance, at node $\langle 1,2,1,2\rangle$ we see that $\pi_2^{\langle 1,2,1,2\rangle}(0) = 1$ while $\pi_2^{\langle 1,2,1,2\rangle}(1) = 3$; this indicates that when $\Delta^X_{R;\langle 1,2,1,2\rangle}(c,(b_0,b_1,b_2,b_3),\vec{a}_0,\vec{a}_1,\vec{a}_2,\vec{a}_3)$ holds, we in particular have $\vec{a}'_0 \subseteq \vec{a}_1$ and $\vec{a}'_1 \subseteq \vec{a}_3$. (The $\rho_i$ functions indicate the corresponding information for the sequences $q_i$.) The ordering of split pairs is significant: the diagram is to be read from left to right in $\prec$-increasing order, so $p_2^+ = q_2^+ \prec p_1^+ = q_1^+$. Finally, the arrows are drawn in a variety of different patterns to make clear which segments are the same; when $\Delta^X_{R;\langle 1,2,1,2\rangle}(c,(b_0,b_1,b_2,b_3),\vec{a}_0,\vec{a}_1,\vec{a}_2,\vec{a}_3)$, the sequence $p_2$ extends the sequence (encoded as part of the datum $b_1$) which was used as $p_2$ to witness that $\Delta^X_{R;\langle 1,2\rangle}(c,(b_0,b_1),\vec{a}_0,\vec{a}_1)$ holds, while $q_2$ extends the sequence (encoded as part of the datum $b_2$) which was identified as $q_1$ when showing that $\Delta^X_{R;\langle 1,2,1\rangle}(c,(b_0,b_1,b_2),\vec{a}_0,\vec{a}_1,\vec{a}_2)$ holds.

We next describe our general construction informally. Each node will be witnessed by a collection of split pairs, say, say $(p_r,q_r),\dots,(p_1,q_1)$ so that $p_{i+1}^+ \prec p_i^+$, each $p_i$ and $q_i$ witnesses some nodes $\sigma_i,\tau_i$ from $T^+$ and $T^-$ respectively, and $q_i$ witnesses a node of length $i$. Some values of $i$ may have

$p_i = q_i = \langle\rangle$; the set $J$ keeps track of those $i$ for which $p_i, q_i$ are non-trivial. We judge progress towards a leaf in $T$ by first looking at how big $r$ is, and then looking at the length of $\sigma_r$, then the length of $\sigma_{r-1}$, and so on.

The general idea is that if $p_r^+ \notin V$ then we can either extend $q_r$ or use it to witness the node $\tau_r$. If $p_{i+1}^+ \in V$ but $p_i^+ \notin V$ then we can either extend $p_{i+1}$ and $q_i$ to some common extension with $(p'_{i+1})^+ = (q'_i)^+$ witnessing extensions of $\sigma_{i+1}$ and $\tau_i$, or we can use $p_{i+1}$ to witness $\sigma_{i+1}$. If $p_1 \in V$ then we can either extend $p_1$ or use it to witness $\sigma_1$.

We now make this precise. Each non-empty node $\upsilon \in T$ will be associated with a non-empty set $J^\upsilon \subseteq [1, \max\{|\tau| \mid \tau \in T^-\}]$, a $j^\upsilon \in J^\upsilon$, and for $j \in J^\upsilon$, non-empty sequences $\sigma_j^\upsilon \in T^+$, $\tau_j^\upsilon \in T^-$, and monotone functions $\pi_j^\upsilon : \mathrm{dom}(\sigma_j^\upsilon) \to \mathrm{dom}(\upsilon)$, $\rho_j^\upsilon : \mathrm{dom}(\tau_j^\upsilon) \to \mathrm{dom}(\upsilon)$. We require that:

- the sets $\mathrm{rng}(\pi_j^\upsilon)$, $\mathrm{rng}(\rho_j^\upsilon)$ are pairwise disjoint except that $\pi_j^\upsilon(|\sigma_j^\upsilon| - 1) = \rho_j^\upsilon(|\tau_j^\upsilon| - 1)$,
- $\pi_{j^\upsilon}^\upsilon(|\sigma_{j^\upsilon}^\upsilon| - 1) = \rho_{j^\upsilon}^\upsilon(|\tau_{j^\upsilon}^\upsilon| - 1) = |\upsilon| - 1$, and
- $|\tau_j^\upsilon| = j$.

Each auxiliary datum $b_i$ will have the form $(e_i, p_i, f_i, q_i)$. For any $j \in J^\upsilon$, let $\hat{p}_j = p_{\pi_j^\upsilon(|\sigma_j^\upsilon|-1)}$ and $\hat{q}_j = q_{\pi_j^\upsilon(|\sigma_j^\upsilon|-1)}$, and let $\hat{p} = \hat{p}_{j^\upsilon} = p_{|\upsilon|-1}$ and $\hat{q} = q_{|\upsilon|-1}$, $\hat{e} = (e_{\pi_{j^\upsilon}^\upsilon(0)}, \dots, e_{\pi_{j^\upsilon}^\upsilon(|\sigma_{j^\upsilon}^\upsilon|-1)})$, and $\hat{f} = (f_{\rho_{j^\upsilon}^\upsilon(0)}, \dots, e_{\rho_{j^\upsilon}^\upsilon(|\tau_{j^\upsilon}^\upsilon|-1)})$. $K_\upsilon^X((b_0, \dots, b_{|\upsilon|-1}), \vec{a})$ will hold if:

- $(\hat{p}, \hat{q})$ is a split pair,
- for each $j \in J^\upsilon$ with $j < j^\upsilon$, $\hat{p}^+ \prec q^+_{\rho_j^\upsilon(|\tau_j^\upsilon|-1)}$,
- for each $j \in J^\upsilon$ with $j > j^\upsilon$, $p^+_{\pi_j^\upsilon(|\sigma_j^\upsilon|-1)} \prec \hat{p}^+$,
- if $j, j' \in J^\upsilon$ with $j < j'$ then $\hat{p}_{j'}^+ \preceq \hat{p}_j^+$,
- if $|\sigma_{j^\upsilon}^\upsilon| > 1$ then $p_{\pi_{j^\upsilon}^\upsilon(|\sigma_{j^\upsilon}^\upsilon|-2)} \sqsubseteq \hat{p}$,
- if $|\sigma_{j^\upsilon}^\upsilon| = 1$ then $p \sqsubseteq \hat{p}$,
- if $|\tau_{j^\upsilon}^\upsilon| > 1$ then $q_{\rho_{j^\upsilon}^\upsilon(|\tau_{j^\upsilon}^\upsilon|-2)} \sqsubseteq \hat{q}$,
- if $|\tau_{j^\upsilon}^\upsilon| = 1$ then $q \sqsubseteq \hat{q}$,
- there are $\vec{a}' \subseteq \vec{a}$ so that $L^{X \oplus \hat{p}}_{\sigma_{j^\upsilon}^\upsilon}(\hat{e}, \vec{a}')$, and
- there are $\vec{a}' \subseteq \vec{a}$ so that $M^{X \oplus \hat{q}}_{\tau_{j^\upsilon}^\upsilon}(\hat{f}, \vec{a}')$.

We can deduce that $d_\upsilon(\pi_j^\upsilon(i)) = d_\sigma^+(i)$ and $d_\upsilon(\rho_j^\upsilon(i)) = d_\tau^-(i)$. This may not fully define $d_\upsilon$, and we may take other values arbitrarily.

Observe the significance of these requirements for $\Delta^X_{R;\upsilon}$: taken collectively, we will have a collection of split pairs $(\hat{p}_j, \hat{q}_j)$ for $j \in J^\upsilon$ so that the endpoints of the $\hat{p}_j$ and $\hat{q}_j$ are in decreasing order and so that each positive requirement component $\Delta^{X \oplus \hat{p}_j}_{R^+;\sigma_j}$ and $\Delta^{X \oplus \hat{q}_j}_{R^-;\tau_j}$ holds (and slightly more—the witnesses to $\Delta^{X \oplus \hat{p}_j}_{R^+;\sigma_j}$ and $\Delta^{X \oplus q_j}_{R^-;\tau_j}$ are chosen from the witnesses to $\Delta^X_{R;\upsilon}$ at suitable levels).

We describe the assignment of these values to $\upsilon$ inductively. We define $J^{\langle\rangle} = \emptyset$. Suppose we have defined $J^\upsilon$ and $\sigma_j^\upsilon, \tau_j^\upsilon, \pi_j^\upsilon, \rho_j^\upsilon$ for each $j \in J^\upsilon$; then we must describe the possible extensions of $\upsilon$. For notational convenience, when $\sigma^v_j$ or $\tau^v_j$ is undefined (because $j \notin J^v$), we take $\sigma^v_j = \tau^v_j = \langle\rangle$ and $\pi^v_j, \rho^v_j$ to be empty functions. We will have extensions of $v$ for each $j \in J^v \cup \{0\}$ such that neither $\sigma^v_{j+1}$ nor $\tau^v_j$ is a leaf.

Fix such a value $j_0$ and chose $\sigma_0$ an immediate extension of $\sigma^v_{j_0+1}$ in $T^+$ and $\tau_0$ an immediate extension of $\tau^v_{j_0}$ in $T^-$. We describe the the various objects needed for the node $v' = v^\frown\langle(j_0, \sigma_0, \tau_0)\rangle$. We take $J^{v'} = (J^v \backslash \{j_0\}) \cup \{j_0 + 1\}$ and $j^{v'} = j_0 + 1$. If $j \in J^{v'} \backslash \{j_0 + 1\}$, we have $\sigma^{v'}_j = \sigma^v_j$, $\tau^{v'}_j = \tau^v_j$, $\pi^{v'}_j = \pi^v_j$, and $\rho^{v'}_j = \rho^v_j$. We take $\sigma^{v'}_{j_0+1} = \sigma_0$ and $\tau^{v'}_{j_0+1} = \tau_0$, $\pi^{v'}_{j_0+1} = \pi^v_{j_0+1} \cup \{(|\sigma_0| - 1, |v'| - 1)\}$ and $\rho^{v'}_{j_0+1} = \rho^v_{j_0} \cup \{(|\tau_0| - 1, |v'| - 1)\}$.

This tree is clearly finitely branching since $T^+, T^-$ are. We should check that it is also finite. To each node $v$, we have the associated pairs $(\sigma^v_j, \tau^v_j)$ for $j \in J^v$. We associate to $v$ a function $v^v : [1, \max\{|\tau| \mid \tau \in T^-] \to [0, \max\{\sigma| \mid \sigma \in T^+\}]$: we define $v^v(j) = |\sigma^v_j|$ if $j \in J^v$ and $v^v(j) = 0$ otherwise. If $v'$ is an immediate extension of $v$ then the only values of $j$ for which $v^{v'}(j) \neq v^v(j)$ are $j^{v'}$ and (possibly) $j^{v'} - 1$; but $v^{v'}(j^{v'}) = v^v(j^{v'}) + 1$. In particular, under the corresponding lexicographic ordering, $v^v$ is strictly increasing and therefore the tree has bounded height.

This completes the specification of our requirement $R$. We may assume that $c$ satisfies $R$ in $X$. Suppose there is an $v$ so that $\Theta^X_{R;v}$ holds. Suppose $v$ is a leaf. First, note that if there is any $j \in J^v$ so that $\sigma^v_j$ and $\tau^v_j$ are both leaves then, since $(\hat{p}_j, \hat{q}_j)$ is a split pair, one of $(\hat{p}_j, q)$ or $(p, \hat{q}_j)$ is a condition. Suppose $(\hat{p}_j, q)$ is a condition; then this condition forces $T^+$ on the increasing side since the leaf $\Theta^{X\oplus\hat{p}_j}_{T^+;\sigma^v_j}(c)$ holds. Similarly, if $(p, \hat{q}_j)$ is a condition then this condition forces $T^-$ on the decreasing side. We claim there is such a $j$. If 0 is not this $j$ then, since $v$ is a leaf, $\sigma^v_1$ must be a leaf. Since $1 \in J^v$ but $\tau^v_1$ is not a leaf, but $v$ is a leaf, we must have $\sigma^v_2$ is a leaf. Continuing in this fashion, each $\sigma^v_j$ is a leaf and each $\tau^v_j$ exists but is not a leaf. But taking $j = \max\{|\tau| \mid \tau \in T^-\}$, we reach a contradiction: $\tau^v_j$ must be a leaf.

Suppose instead $v$ is not a leaf. Taking $\hat{q}_0 = q$, let $j \in J^v \cup \{0\}$ be largest such $\hat{q}^+_j \notin V$. If $\tau^v_j$ is a leaf (so $j \neq 0$) then $(p, \hat{q}_j)$ is an extension and $\Theta^{X\oplus\hat{q}}_{T^-;\tau^v_j}$ holds, so we are done. So assume $\tau^v_j$ is not a leaf. If $j + 1 \in J^v$, let $\hat{p} = \hat{p}_{j+1}$, otherwise let $\hat{p} = p$. Observe that $\hat{p}^+ \in V$: if $\hat{p} \neq p$ then $\hat{p} = \hat{p}_{j+1}$ and $\hat{p}^+_{j+1} = \hat{q}^+_{j+1} \in V$ by choice of $j$. If $\sigma^v_{j+1}$ is a leaf then, similarly, $(\hat{p}, q)$ is an extension and we are done.

So suppose neither $\sigma^v_{j+1}$ nor $\tau^v_j$ are leaves. $(\hat{p}, \hat{q})$ is an extension which we claim forces either $T^+$ on the increasing side or $T^-$ on the decreasing side. For suppose not, so there are $\Lambda^+$ and $\Lambda^-$ witnessing this failure. In particular, since $\Delta^{X\oplus\Lambda^+}_{R^+;\sigma^v_{j+1}}$ and $\Delta^{X\oplus\Lambda^-}_{R^-;\tau^v_j}$ hold but $\Theta^{X\oplus\Lambda^+}_{R^+;\sigma^v_{j+1}}$ and $\Theta^{X\oplus\Lambda^-}_{R^-;\tau^v_j}$ do not, there must be finite $p^*, q^*$ so that $\hat{p} \sqsubseteq p^* \sqsubset \Lambda^+$ and $\hat{q} \sqsubseteq q^* \sqsubset \Lambda^-$ large enough to witness that our chosen witnesses to the positive requirements do not succeed in witnessing the negative requirements. If $j + 1 \notin J^v$, so $\hat{p} = p$, we might have some $j' > j$ with $j' \in J^v$ and $\hat{p}^+_{j'} \not\prec (p^*)^+$. In this case we replace $p^*$ by $p^*\frown\langle \hat{p}^+_{j'}\rangle$. We may extend $q^*$ by a single element so that $(p^*)^+ = (q^*)^+$. Then $(p^*, q^*)$ is a split pair witnessing the failure of $\Theta^X_{R;\upsilon}$, contradicting the assumption. □

2.3. **Solving CAC.** It is convenient to restrict ourselves to partial orderings which are refinements of the usual ordering on $<$; the following lemma shows that this restriction is harmless for our purposes.

**Lemma 2.12.** *Suppose $\mathcal{I}$ is a Turing ideal and whenever $\preceq$ is a partial ordering in $\mathcal{I}$ so that $a < b$ implies $b \npreceq a$, $\mathcal{I}$ contains either an infinite chain or an infinite chain in $\preceq$. Then $\mathcal{I}$ contains an infinite chain or antichain for every partial ordering.*

*Proof.* Let $\preceq$ be an arbitrary partial ordering in $\mathcal{I}$. Define $a \preceq' b$ if $a \leqslant b$ and $a \preceq b$. Then $\mathcal{I}$ contains either a chain or an antichain for $\preceq'$; if $\mathcal{I}$ contains a chain then it is also a chain in $\preceq$. Suppose $n_1 < n_2 < \cdots$ is an infinite antichain in $\mathcal{I}$. For $a \leqslant b$, define $a \preceq^* b$ if $n_b \preceq n_a$. Then $\preceq^*$ is a partial ordering with a chain or an antichain in $\mathcal{I}$, which is also a chain or antichain for $\preceq$. □

**Lemma 2.13.** *Suppose $c$ satisfies every requirement in $X$ and $\preceq$ is a partial ordering so that $a < b$ implies $b \npreceq a$. Then there is an infinite $\Lambda$ which is either a chain or an antichain so that $c$ satisfies every requirement in $X \oplus \Lambda$.*

We attempt to outline the construction before the proof. First, we note some general features of constructions. (These features were already present in the **ADS** construction above, which illustrates these ideas more concretely.) Say we have some requirements $R^+$ and $R^-$, and we are attempting to produce a requirement $R$ so that whenever some $\Theta^X_{R;\upsilon}(c)$ holds, we have either a chain $p$ so that some $\Theta^{X\oplus p}_{R^+;\sigma}(c)$ holds or some antichain $q$ so that some $\Theta^{X\oplus q}_{R^-;\tau}(c)$ holds.

When $\Delta^X_{R;\upsilon}(c, b_0, \ldots, b_{|\upsilon|-1}, \vec{a}_0, \ldots, \vec{a}_{|\upsilon|-1})$ holds, the $b_i$ must encode the description of a list of chains $p_1, \ldots, {}_{r^+}$ and antichains $q_1, \ldots, q_{r^-}$ which are candidates to be the needed witnesses; the $b_i$ must also encode the corresponding nodes $\sigma_i \in T^+$ and $\tau_i \in T^-$ which these chains or antichains would witness. (We will call these the "witnessing chains" and "witnessing antichains", respectively.)

Consider some candidate chain $p_i$. This chain must have been constructed in $|\sigma_i|$ segments, with each segment corresponding to some stage of $\upsilon$: that is, there should be a function $\pi_i : [0, |\sigma_i|) \to [0, |\upsilon|)$ so that when $\Delta^X_{R;\upsilon}(c, b_0, \ldots, b_{|\upsilon|-1}, \vec{a}_0, \ldots, \vec{a}_{|\upsilon|-1})$ holds, $\Delta^{X\oplus p_i}_{R^+;\sigma_i}(c, b'_{\pi_i(0)}, \ldots, b'_{\pi_i(|\sigma_i|-1)}, \vec{a}'_{\pi_i(0)}, \ldots, \vec{a}'_{\pi_i(|\sigma_i|-1)})$ where $\vec{a}'_i \subseteq \vec{a}_i$ and each $b'_i$ is encoded in $b_i$. (The functions $\pi_i$, the corresponding functions for antichains, which we denote with variants on the letter $\rho$, are encoded in the structure of $\upsilon$.)

We can make a crucial observation about the stages at which our witnessing chains and antichains get extended. Suppose that $\upsilon'$ is some immediate successor of $\upsilon$, and that there is a witnessing chain $p'$ at stage $\upsilon'$ with

$\pi'(|\sigma'|-1)=|\upsilon|$—that is, at stage $\upsilon$ there was a witnessing chain $p$ and $p'$ is a proper immediate extension of it (so $\upsilon$ was one of the stages at which $p'$ was extended). Then we must have had $d_\upsilon(\pi(i))=d_\sigma(i)$ for all $i<|\sigma|$. When this happens, we say $p$ is *active* at $\upsilon$; otherwise we say $p$ is *inactive*.

This basic structure, of active and inactive witnessing chains and antichains (or their generalizations) constructed in stages and the functions $\pi$ and $\rho$ which correspond stages of $\upsilon$ with stages of $\sigma$ or $\tau$, will appear in all our arguments.

With that in mind, we turn to the simplest case for **CAC**. We have two requirements of length 1, say $R^+$ and $R^-$. For simplicity, let us say $T^+$ and $T^-$ each consist of a single non-empty node with simple block statements $K_+$ and $K_-$. We wish to do one of the following:

- find a chain $p$ so that there are infinitely many $s \succ p^+$, and there exist $b,\vec{a}$ so that $\Delta_+^{X\oplus p}(b,\vec{a})$ holds,
- find an antichain $q$ so that there are infinitely many $s$ incomparable to every element of $q$, and there exist $b,\vec{a}$ so that $\Delta_-^{X\oplus q}(b,\vec{a})$ holds,
- find an infinite $X$-computable set $S$ so that if $p\subseteq S$ is a chain then there are no $b,\vec{a}$ satisfying $\Delta_+^{X\oplus p}(b,\vec{a})$, or
- find an infinite $X$-computable set $S$ so that if $q\subseteq S$ is an antichain then there are no $b,\vec{a}$ satisfying $\Delta_-^{X\oplus q}(b,\vec{a})$.

Moreover, we need to express this construction in the form of a requirement $R$ so that we can use the fact that $c$ satisfies all requirements in $X$ to ensure that if there is such a chain or antichain, there is one where the witnesses belong to the appropriate sets.

One possibility is that we can, after throwing away finitely many elements, rule out the possibility of finding a chain: if there is some $t$ so that no witnessing chain has $t<p^+$, then we could simply take $S=(t,\infty)$ and would have satisfied the third case.

So suppose not: above any $t$ we can find a witnessing chain. In particular, we can find an infinite computable set $S$ consisting entirely of the endpoints of witnessing chains. If this set witnesses the fourth case, we are done.

If not, we can find a witnessing antichain $q$ so that, for each $x\in q$, $x=p^+$ for some witnessing chain $p$. Our first block of witnessing elements include all the $\vec{a}$ witnessing $q$, as well as those witnessing all the $p$ whose endpoints make up $q$. Finding such a set $q$ is the first node in our requirement $R$; upon finding it, we apply its restraint rule to the witnesses in this block.

If there are infinitely many $y$ which are incomparable to every element of $q$ then we have satisfied the second case. If not, cofinitely many elements are above some $x\in q$. In particular, there must be an $x\in q$ so that the set of $y$ with $x\prec y$ is infinite. If this set witnesses the fourth condition, we are again done.

Otherwise, for one of our witnessing chains $p$, we find a witnessing antichain $q'$ with $p^+\prec y$ for each $y\in q'$. This is our second node; the witnesses for $p$ belong to the first block, and the witnesses for $q'$ to the second block.

Then every $z$ is either incomparable to every element of $q'$, or there is some $y \in q'$ with $y \prec z$, and since $p^+ \prec z$, we have $p^+ \prec z$ as well. Therefore either $p$ witnesses the first case or $q'$ witnesses the second case. This completes the construction of $R$.

We note that there are two distinct attempts to find an antichain; these form a key part of our construction, so we give them names. The first attempt, when we construct an antichain $q$ out of endpoints of chains, we call a *trial antichain*. The second attempt, when we find $q'$ above some chain $p$, we call a *partnered antichain* (because, unlike a trial antichain, it is partnered with the chain $p$ which is active at the same time $q'$ is).

We now consider the case where $T^+ = T^- = \{\langle\rangle, \langle 0\rangle, \langle 0, 0\rangle\}$.

In Figure 2, we represent the tree of nodes corresponding to a single construction of a trial solution. Note that these nodes might not be adjacent in the broader tree—the actual tree contains many copies of these nodes, allowing the possibility that other constructions act to extend other trial solutions between adjacent steps on this trial solution. We indicate above each node how it will be labeled; trial solutions are indexed by functions $\omega$, so the common choice of $\omega$ indicates that these nodes all correspond to the same trial solution.

The horizontal lines indicate antichains, the vertical arrows represent chains. The doubled lines indicate that the chain or antichain is active. Initially, in the node indexed by $(0, \omega, \sigma_0)$, we discover an antichain whose elements are endpoints of chains; the antichain witnesses a node $\sigma_0 \in T^-$, while the chains $p_0^1, \dots, p_0^{k_0}$ witness nodes in $T^+$. One possible extension, the node $(1, \omega, \sigma^0, \tau^0, 0)$, occurs if we find a new witnessing chain $q$ entirely above $p_0^k$ for some $k \leqslant k_0$; this node completes the construction of this trial solution, since we have found a partnered antichain.

In the other path, $(0, \omega, \sigma_1)$ we find an extension of the antichain $\hat{q}_0$ to an antichain $\hat{q}_1$, witnessing some $\sigma_1$ extending $\sigma_0$, where the elements of $\hat{q}_1$ are the endpoints of a new family of chains, $p_1^1, \dots, p_1^{k_1}$. Since $T^-$ has no nodes of length larger than 2, $\hat{q}_1$ must witness a leaf from $T^-$, so we have completed its construction. The remaining possibility is that we find a partnered antichain; there are two versions, depending on whether the antichain is above one of the chains from $\hat{q}_0$ or one of the chains from $\hat{q}_1$.

The technique of using the trial solutions can be isolated as the following black box property:

> let $p$ be a chain witnessing a stage $\sigma \in T^+$ and $q$ an antichain witnessing a stage $\tau \in T^-$; then either:
> - there is an antichain $\hat{q}$ witnessing $R^-$, or
> - there are extensions $p' \sqsupset p$ and $q' \sqsupset q$ witnessing a child $\sigma'$ or $\sigma$ and $\tau'$ of $\tau$ respectively, and so that $(p')^+ \prec q'$.

In our larger construction, we apply this subconstruction repeatedly. We should imagine we are attempting to make an antichain and, for each segment of this antichain, a perpindicular chain. This is illustrated in Figure 3; each

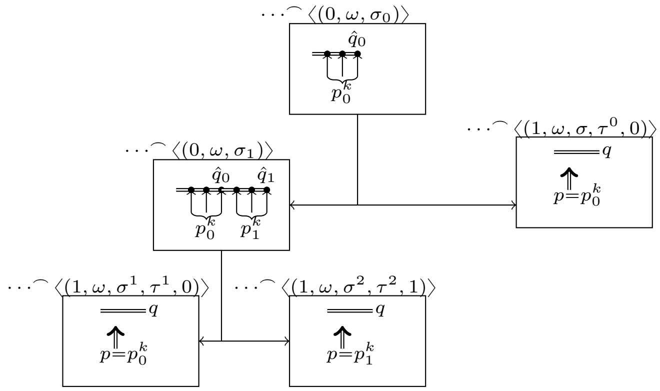


FIGURE 2.

solid box represents a stage of the higher level construction, and each dotted box represents a trial solution which may be active at that stage. Initially there is one trial solution, working on the first segment of the antichain and a chain of length one partnered with it; once we find that, we can either work on extending the length of the chain (which means replacing the antichain segment with a new one) or on extending the antichain (by partnering the second segment with a corresponding antichain).

Note that there are many nodes in our actual requirement between two boxes in this picture; for instance, when there are multiple active trial solutions, each requires several steps, and steps from different trial solutions could be interspersed on our actual requirement; therefore each stage in this high level diagram corresponds to many nodes in a requirement. Also, the stages on the right side can also extend the first chain, causing them to reset progress on the second antichain; in our actual requirement, these still have to lead to separate nodes (because a requirement is a tree).

There is a further complication which is hidden in Figure 3, but which we will encounter again. Consider the first stage on the right side, where both the first and second antichains have length one. On the one hand, we need the two segments of antichain to, together, correspond to a node of length 2 from $T^-$. On the other hand, we need it to be possible to extend the first segment of the antichain with a different second segment as represented in the block below it. This is not actually possible to do with a single witnessing antichain simultaneously (the two nodes in $T^-$ they correspond to may call for different requirements which cannot be active simultaneously). So the dotted rectangle in the lower left must actually correspond to two different partnered antichains: in the first stage on the right, one of them has been extended to a second segment, while the other is waiting for an extension, and in the second stage on the right, the former has been discarded while the second has found its extension.

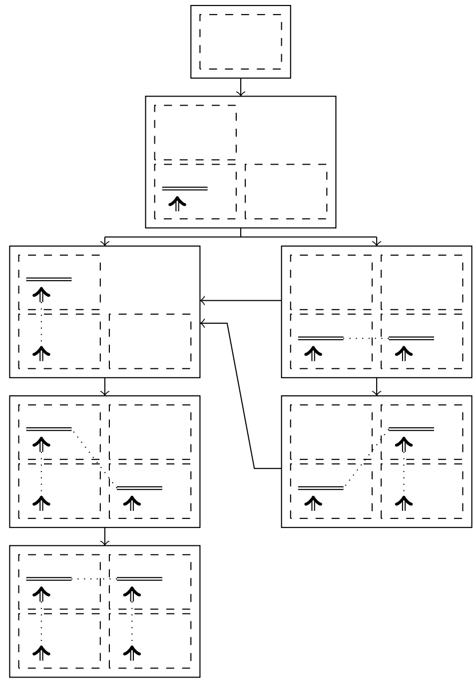

FIGURE 3.

In order to make the proof below more intelligible, we now consider the expansion of the higher level construction into a more explicit one; in particular, we name the various chains and antichains and keep track of the multiple copies corresponding to some positions. We need functions $\omega : [0, 2) \to (0, 2]$; we write these as sequences: $\omega = \langle 1, 2\rangle$ indicates the function with $\omega(0) = 1$ and $\omega(1) = 2$. We also need functions $\gamma$ which are terminal subsequences; in this case there are only three possible values for $\gamma$: $\langle 1\rangle$, $\langle 2\rangle$, and $\langle\rangle$.

In Figure 4, each solid box represents a node of our tree. Within each node, we have several constructions of trial solutions, indicated by boxes with dotted lines. The boxes are labeled by an $\omega$, indicated in the lower right corner, and a $\gamma$, indicated in the lower right. (Roughly speaking, $\omega$ names the antichain while $\gamma$ names the chain.) We ignore the internal structure of the trial solutions: this diagram only indicates those nodes at which a partnering happens within some trial solution.

Because there is a large amount of branching, we only consider one path through our tree, which runs down the first column and then wraps around to the second (one of the longer paths, so we can illustrate the main features of the construction). When one portion is directly to the right of another portion, we expect the thing on the right to be incomparable to the thing on the left. When one portion is directly above another portion, we expect it to be above (in the sense of $\prec$) the lower portion.

Between the fifth and sixth nodes (i.e. between the bottom of the first column and the top of the second) and again between the seventh and eigth, we see a trial solution corresponding to a longer $\gamma$ extend, and notice that this causes us to discard all trial solutions coming from the shorter $\gamma$. (In fact, in this case, neither discard was necessary—a more efficient tree would have kept the discarded portion in both cases. However in other situations this discard is necessary, and the indexing is simpler if we do it indiscriminately.)

*Proof.* We force with conditions which are triples $(p, q, S)$ so that:

- $p$ is a chain,
- $q$ is an antichain,
- $S$ is an infinite $X$-computable set, $p < S$, $q < S$, if $a \in p$, $b \in q$, and $c \in S$ then $a \prec c$ and $b \nprec c$.

A condition $(p', q', S')$ extends $(p, q, S)$ if $p \sqsubseteq p'$, $q \sqsubseteq q'$, $(p' \backslash p) \subseteq S$, $(q' \backslash q) \subseteq S$, and $S' \subseteq S$. We say $(p, q, S)$ *forces $R$ on the chain side* if whenever $\Lambda$ is an infinite chain extending $p$ with $\Lambda \backslash p \subseteq S$, $c$ satisfies $R$ in $X \oplus \Lambda$. Similarly, we say $(p, q, S)$ *forces $R$ on the antichain side* if whenever $\Lambda$ is an infinite antichain extending $q$ with $\Lambda \backslash q \subseteq S$, $c$ satisfies $R$ in $X \oplus \Lambda$.

For any $x \in S$, let $S_{\succ x} = \{y \in S \mid x \prec y\}$ and $S_{\perp x} = \{y \in S \mid x < y, x \nprec y\}$, so $S \backslash (S_{\succ x} \cup S_{\perp x})$ is finite. Then either $(p^\frown \langle x \rangle, q, S_{\succ x})$ or $(p, q^\frown \langle x \rangle, S_{\perp x})$ is a condition. In particular, we may always extend at least one of $p$ and $q$ by one element. Furthermore, if there do not exist at least one $x$ which can be added to the $p$ side and at least one which can be added to the $q$ side then $\preceq$ has an $X$-computable chain or antichain: say there is no $x$ which can be added to the $q$ side, so for every $x \in S$, $S_{\perp x}$ is finite. Then we can greedily add elements from $S$ to $p$ and obtain an infinite chain.

So it suffices to show:

> $(*)$ Suppose $R^+$ and $R^-$ are requirements and $(p, q, S)$ is a condition. Then there is a condition $(p', q', S')$ extending $(p, q, S)$ which either forces $R^+$ on the chain side or $R^-$ on the antichain side.

For suppose we have shown this. Then we fix a list of requirements $R_i^+, R_i^-$ so that for any pair of requirements $R^+, R^-$, there is an $i$ with $R_i^+ = R^+, R_i^- = R^-$. We construct a sequence $(\langle\rangle, \langle\rangle) = (p_0, q_0, S_0), (p_1, q_1, S_1), \ldots$ with $(p_{i+1}, q_{i+1}, S_{i+1})$ extends $(p_i, q_i, S_i)$, $(p_{2i+1}, q_{2i+1}, S_{2i+1})$ either forces $R_i^+$ on the chain side or $R_i^-$ on the antichain side, $p_{2i}$ has length $\geqslant i$, and $q_{2i}$ has length $\geqslant i$. Let $\Lambda^+ = \bigcup p_i$ and $\Lambda^- = \bigcup q_i$. If $c$ does not satisfy every requirement in $X \oplus \Lambda^+$ then there is some $R^+$ which it fails to satisfy, and therefore for each $R^-$ there was an $i$ with $R_i^+ = R^+, R_i^- = R^-$, and therefore since $(p_{2i+1}, q_{2i+1})$ must not have forced $R^+$ on the chain, $(p_{2i+1}, q_{2i+1})$ forced $R^-$ on the antichain, and therefore $\Lambda^-$ satisfies every requirement in $X \oplus \Lambda^-$.

So it suffices to show $(*)$. Let $R^+ = (T, \{L_\sigma\}_{\sigma \in T^+}, \{d_\sigma^+\}_{\sigma \in T^+})$ and $R^- = (T^-, \{M_\tau\}_{\tau \in T^-}, \{d_\tau^-\}_{\tau \in T^-})$. Let $D = \max\{|\sigma| \mid \sigma \in T^+\}$ and $E = \max\{|\tau| \mid \tau \in T^-\}$. We will describe a requirement $R = (T, \{K_v\}_{v \in T}, \{d_v\}_{v \in T})$.

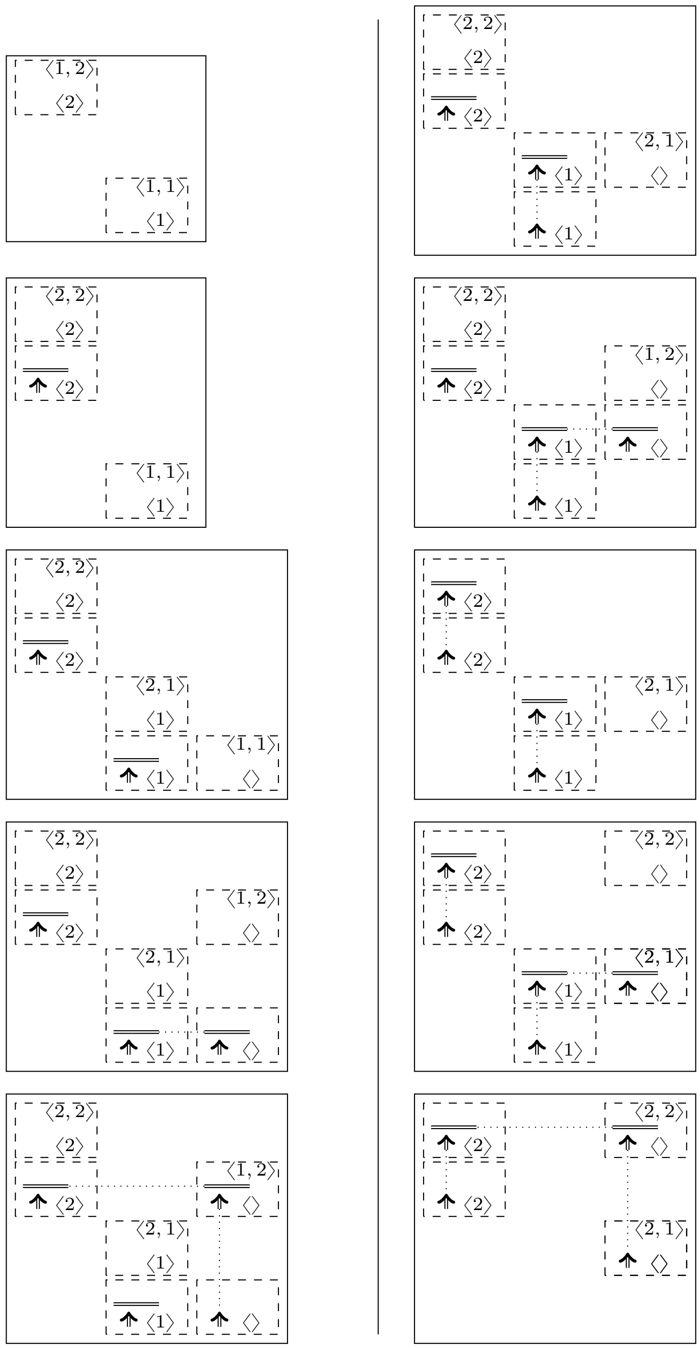

FIGURE 4.

The first thing node $\upsilon \in T$ must keep track of is the various partnered antichains which have been constructed, and their partner chains. A partnered antichain is indexed by a function $\omega : [0, E) \to (0, D]$ (the value $\omega(i)$ indicates the length of the chain supporting the $i$-th segment of the antichain); the corresponding chains are indexed by terminal partial subfunctions: let $G$ be the collection of functions $\gamma : (r_\gamma, E) \to (0, D]$ with $r_\gamma \in [0, E)$.

Each node $\upsilon \in T$ will be associated function $\theta^\upsilon : G \to [0, D]$ indiating the length of the chain we have constructed for $\gamma$. This is enforced by requiring that, for each $\gamma \in G$, we have:

- a $\sigma^\upsilon_\gamma \in T^+$ with $|\sigma^\upsilon_\gamma| = \theta^\upsilon(\gamma)$, and
- a monotone function $\pi^\upsilon_\gamma : \mathrm{dom}(\sigma^\upsilon_\gamma) \to \mathrm{dom}(\upsilon)$.

$\omega$ is *allowed* at $\upsilon$ if for each $E' < E$, $\theta^\upsilon(\omega \upharpoonright (E', E)) \leqslant \omega(E')$. (Otherwise we have already grown some chains long enough that we no longer need to consider the potential partnered antichain indexed by $\omega$.) When $\omega$ is allowed, $E^\upsilon_\omega \in [0, E)$ is the largest value so that, for each $E' \in [0, E^\upsilon_\omega)$, and $\omega(E') = \theta^\upsilon(\omega \upharpoonright (E', E))$. (Note that $E^\upsilon_\omega$ may be 0.)

For each allowed $\omega$, we have:

- a $\tau^\upsilon_\omega \in T^-$ with $|\tau^\upsilon_\omega| = E^\upsilon_\omega$,
- a monotone function $\rho^\upsilon_\omega : \mathrm{dom}(\tau^\upsilon_\omega) \to \mathrm{dom}(\upsilon)$.

$\omega$ is *active* at $\upsilon$ if $E^\upsilon_\omega < E - 1$ and $\omega(E^\upsilon_\omega) = \theta^\upsilon(\omega \upharpoonright (E^\upsilon_\omega + 1, E)) + 1$. For each active $\omega$ we also need a trial antichain, represented by:

- a $\hat{\tau}^\upsilon_\omega \in T^-$,
- $\hat{\rho}^\upsilon_\omega : \mathrm{dom}(\tau^\upsilon_\omega) \to \mathrm{dom}(\upsilon)$.

We require that, across all $\omega$ and $\gamma$, the sets $\mathrm{rng}(\pi^\upsilon_\gamma), \mathrm{rng}(\rho^\upsilon_\omega)$, and $\mathrm{rng}(\hat{\rho}^\upsilon_\omega)$ are pairwise disjoint.

We can deduce that $d_\upsilon(\rho^\upsilon_\omega(j)) = d^-_{\tau^\upsilon_\omega}(j)$, $d_\upsilon(\hat{\rho}^\upsilon_\omega(j)) = d^-_{\hat{\tau}^\upsilon_\omega}(j)$, and $d_\upsilon(\pi^{\upsilon,i}_\omega(j)) = d^+_{\sigma^{\upsilon,i}_\omega}(j)$. This may not fully define $d_\upsilon$, and we may take other values to be arbitrary.

Each non-empty node $\upsilon$ belongs to one of two types, depending on whether it represents the extension of a trial antichain or of a partnered antichain. When there is some active $\omega$ so that $\hat{\rho}^\upsilon_\omega(|\hat{\tau}^\upsilon_\omega| - 1) = |\upsilon| - 1$, we call $\upsilon$ a *trial node*. When there is some active $\omega$ with $\rho^\upsilon_\omega(|\tau^\upsilon_\omega| - 1) = |\upsilon| - 1$, we call $\upsilon$ a *partnering node*.

We describe the assignment of these values to $\upsilon$ inductively. When $\upsilon = \langle\rangle$, $\theta^\upsilon(\gamma) = 0$ and $\sigma^\upsilon_\gamma = \langle\rangle$ for all $\gamma \in G$, $\tau^\upsilon_\omega = \langle\rangle$ for all allowed $\omega$, and $\hat{\tau}^\upsilon_\omega = \langle\rangle$ for all active $\omega$.

Suppose we have defined everything for some node $\upsilon$; we describe possible extensions of $\upsilon$. For each active $\omega_0$ and each immediate extension $\tau$ of $\hat{\tau}^\upsilon_{\omega_0}$ in $T^-$, we have an extension $\upsilon' = \upsilon^\frown\langle(0, \omega_0, \tau)\rangle$, a trial node, with:

- $\theta^{\upsilon'} = \theta^\upsilon$ and for each $\gamma$, $\sigma^{\upsilon'}_\gamma = \sigma^\upsilon_\gamma$, and $\pi^{\upsilon'}_\gamma = \pi^\upsilon_\gamma$,
- for each allowed $\omega$, $\tau^{\upsilon'}_\omega = \tau^\upsilon_\omega$ and $\rho^{\upsilon'}_\omega = \rho^\upsilon_\omega$,
- $\hat{\tau}^{\upsilon'}_{\omega_0} = \tau$ and $\hat{\rho}^{\upsilon'}_{\omega_0} = \hat{\rho}^\upsilon_{\omega_0} \cup \{(|\tau| - 1, |\upsilon'| - 1)\}$,

- for $\omega \neq \omega_0$, $\hat{\tau}^{\upsilon'}_\omega = \hat{\tau}^\upsilon_\omega$ and $\hat{\rho}^{\upsilon'}_\omega = \hat{\rho}^\upsilon_\omega$.

For each active $\omega_0$, let $\gamma_0 = \omega_0 \upharpoonright (E^\upsilon_{\omega_0}, E)$. For each $\sigma$ an immediate extension of $\sigma^\upsilon_{\gamma_0}$, each $\tau$ an immediate extension of $\tau^\upsilon_{\omega_0}$, and each $j_0 < |\hat{\tau}^\upsilon_{\omega_0}|$, there is an extension $\upsilon' = \upsilon^\frown\langle(1, \omega_0, \sigma, \tau, j_0)\rangle$, a partnering node, with:

- if $r_\gamma > r_{\gamma_0}$, $\theta^{\upsilon'}(\gamma) = 0$, $\sigma^\upsilon_\gamma = \langle\rangle$, and $\pi^\upsilon_\gamma = \emptyset$,
- $\theta^{\upsilon'}(\gamma_0) = \theta^\upsilon(\gamma_0) + 1$, $\sigma^{\upsilon'}_{\gamma_0} = \sigma$, and $\pi^{\upsilon'}_{\gamma_0} = \pi^\upsilon_\gamma \cup \{(|\sigma^\upsilon_{\gamma_0}|, \hat{\rho}^\upsilon_{\omega_0}(j_0))\}$,
- if $r_\gamma \leqslant r_{\gamma_0}$ and $\gamma \neq \gamma_0$ then $\theta^{\upsilon'}(\gamma) = \theta^\upsilon(\gamma)$, $\sigma^{\upsilon'}_\gamma = \sigma^\upsilon_\gamma$, or $\pi^{\upsilon'}_\gamma = \pi^\upsilon_\gamma$,
- $\tau^{\upsilon'}_{\omega_0} = \tau$ and $\rho^{\upsilon'}_{\omega_0} = \rho^\upsilon_\omega \cup \{(|\tau^\upsilon_{\omega_0}|, |\upsilon|)\}$,
- for each $\omega \neq \omega_0$, $E^{\upsilon'}_\omega = \min\{r_{\gamma_0}, E^\upsilon_\omega\}$, $\tau^{\upsilon'}_\omega = \tau^\upsilon_\omega \upharpoonright E^{\upsilon'}_\omega$, and $\rho^{\upsilon'}_\omega = \rho^\upsilon_\omega \upharpoonright E^{\upsilon'}_\omega$,
- for each $\omega$, $\hat{\tau}^{\upsilon'}_\omega = \langle\rangle$ and $\hat{\rho}^{\upsilon'}_\omega = \emptyset$.

The tree is finitely branching since $T^+$ and $T^-$ are. To see that it is finite, we first observe that there cannot be infinitely many consecutive trial nodes: there are finitely many $\omega_0$, each of which can belong to at most $E$ trial nodes in a sequence of consecutive nodes containing no partnering nodes. So it suffices to show that there cannot be infinitely many partnering nodes on a path: the $\gamma$ with $r_\gamma = 0$ can only participate in $D$ nodes (since each partnering node extends $\sigma^\upsilon_\gamma$, and when $r_\gamma = 0$, $\sigma^\upsilon_\gamma$ is never reset); after the last partnering node for a $\gamma$ with $r_\gamma = 0$, each $\gamma$ with $r_\gamma = 1$ can only participate in $D$ more nodes, and so on.

When $\upsilon = \upsilon_0{}^\frown\langle(0, \omega_0, \tau)\rangle$ is a trial node, let $\gamma_0 = \omega_0 \upharpoonright (E^\upsilon_{\omega_0}, E)$ and $\ell = |\upsilon| - 1$. The auxiliary datum $b_\ell$ has the form $(k_\ell, p^0_\ell, e^0_\ell, \ldots, p^{k_\ell}_\ell, e^{k_\ell}_\ell, q_\ell, f_\ell)$. Let $\sigma_0 = \sigma^\upsilon_{\gamma_0}$ and let $p' = p$ if $\theta^\upsilon(\gamma_0) = 0$ and $p' = p_{\pi^\upsilon_{\gamma_0}(|\sigma^\upsilon_{\gamma_0}|-1)}$ otherwise Let $q' = q$ if $|\hat{\tau}^{\upsilon_0}_{\omega_0}| = 0$ and $q' = q_{\hat{\rho}^\upsilon_{\omega_0}(|\hat{\tau}^{\upsilon_0}_{\omega_0}|-1)}$ otherwise. $K^X_\upsilon((b_0, \ldots, b_\ell), \vec{a})$ will hold if:

- for each $k \leqslant k_\ell$, $p' \sqsubseteq p^k_\ell$ and $p^k_\ell \subseteq S$,
- for each $k \leqslant k_\ell$, $p^k_i$ is a chain so that there is an $\vec{a}' \subseteq \vec{a}$ and some $\sigma^k_\ell$ immediately extending $\sigma_0$ so that $L^{X \oplus p^k_\ell}_{\sigma^k_\ell}((e_{\pi^\upsilon_{\gamma_0}(0)}, \ldots, e_{\pi^\upsilon_{\gamma_0}(|\sigma^\upsilon_{\gamma_0}|-1)}, e^k_\ell), \vec{a}')$ holds,
- $q' \sqsubseteq q_\ell$,
- every element of $q_\ell \backslash q'$ is an endpoint of $p^k_\ell$ for some $k \leqslant k_\ell$ (and therefore also contained in $S$),
- $q_\ell$ is an antichain so that there is $\vec{a}' \subseteq \vec{a}$ so that $M^{X \oplus q_\ell}_\tau((f_{\hat{\rho}^\upsilon_{\omega_0}(0)}, \ldots, f_{\hat{\rho}^\upsilon_{\omega_0}(|\hat{\tau}^\upsilon_{\omega_0}|-1)}, f_\ell), \vec{a}')$ holds.

When $\upsilon = \upsilon_0{}^\frown\langle(1, \omega_0, \sigma, \tau, j_0)\rangle$ is a partnering node, let $\gamma_0 = \omega_0 \upharpoonright (E^\upsilon_{\omega_0}, E)$ and $\ell = |\upsilon| - 1$. The auxiliary datum $b_\ell$ has the form $(p_\ell, q_\ell, f_\ell)$. If $|\tau^{\upsilon_0}_{\omega_0}| = 0$ let $q' = q$, otherwise let $q' = q_{\rho^{\upsilon_0}_{\omega_0}(|\tau^{\upsilon_0}_{\omega_0}|-1)}$. Then $K^X_\upsilon((b_0, \ldots, b_\ell), \vec{a})$ holds if:

- $p_\ell = p^k_{\hat{\rho}^{\upsilon_0}_{\omega_0}(j_0)}$ for some $k \leqslant k_{\hat{\rho}^{\upsilon_0}_{\omega_0}(j_0)}$ with $\sigma = \sigma^k_{\hat{\rho}^{\upsilon_0}_{\omega_0}(j_0)}$,
- $q' \sqsubseteq q_\ell$,
- every element of $q_\ell \backslash q'$ is above the endpoint of $p_\ell$ and contained in $S$,

- there is some $\vec{a}' \subseteq \vec{a}$ so that $M_\tau^{X\oplus q_\ell}((f_{\rho^\upsilon_{\omega_0}(0)}, \dots, f_{\rho^\upsilon_{\omega_0}(|\tau^\upsilon_{\omega_0}|-1)}, f_\ell), \vec{a}')$ holds.

By assumption, there must be some $\upsilon \in T$ so that $\Theta^X_{R;\upsilon}(c)$ holds. We now show that we can find the needed extension of $(p, q, S)$. The main complication is identifying which of our many potential chains and antichains is the right one to use. We first rule out the case where some $\gamma$ or some $\omega$ is easily identifiable as the correct extension.

First, suppose there is any $\gamma$ so that $\sigma^\upsilon_\gamma$ is a leaf and $S' = S_{\prec (p^\upsilon_{\pi^\upsilon_\gamma(|\sigma^\upsilon_\gamma|-1)})^+}$ infinite. Then $(p^\upsilon_{\pi^\upsilon_\gamma(|\sigma^\upsilon_\gamma|-1)}, q, S')$ is the desired extension, and we are done.

Next, suppose there is an $\omega$ so that $\tau^\upsilon_\omega$ is a leaf and, for each $E' \leqslant E^\upsilon_\omega$, taking $\gamma_{E'} = \omega \upharpoonright (E', E)$, $\sigma^\upsilon_{\gamma_{E'}}$ is a leaf. Let $q' = q_{\rho^\upsilon_\omega(|\tau^\upsilon_\omega|-1)}$; for each $x \in q'$, there is an $E'$ so that we have $x \succ p^\upsilon_{\pi^\upsilon_{\gamma_{E'}}(|\sigma^\upsilon_{\gamma_{E'}}|-1)}$, and therefore, by assumption, $S_{<x}$ is finite. Therefore $S' = \bigcap_{x\in q'} S_{\perp x}$ is infinite, and therefore $(p, q', S')$ is the desired extension.

So we further suppose there is no such $\omega$. For each $\gamma$, take $p_\gamma = p$ if $\theta^\upsilon(\gamma) = 0$ and $p_\gamma = p^\upsilon_{\pi^\upsilon_\gamma(|\sigma^\upsilon_\gamma|-1)}$ otherwise. Suppose that there is no $\gamma$ with $S_\prec = S_{\prec p^+_\gamma}$ infinite. Note that this implies that there is no $\gamma$ with $\theta^\upsilon(\gamma) = 0$. Let $\gamma_E = \langle\rangle$; given $\gamma_{E'}$, define $\gamma_{E'-1} = \langle \theta^\upsilon(\gamma_{E'})\rangle^\frown \gamma_{E'}$. Let $\omega = \gamma_{-1}$. Then $\omega$ is allowed and $\sigma^\upsilon_\omega$ is a leaf. Take $q' = q_{\rho^\upsilon_\omega(E-1)}$. For each $x \in q'$, $x \succ p_{\gamma_{E'}}$ for some $E'$, so and since $S_{<x}$ is finite, $\bigcap_{x\in q'} S_{\perp x}$ is infinite. Therefore $(p, q', \bigcap_{x\in Q'} S_{\perp x})$ is the desired extension.

We now to look at active $\omega$ and the corresponding $\gamma = \omega \upharpoonright (E^\upsilon_\omega, E)$, as well as the corresponding trial solution. We need to pick out which of the active $\omega$ is the right one to examine.

Choose some $\gamma$ so that $r_\gamma$ is least among those $\gamma$ with $S_\prec = S_{<p^+_\gamma}$ is infinite. Note that for any $\gamma'$ with $r_{\gamma'} < r_\gamma$, $\theta^\upsilon(\gamma') > 0$. Further, note that $\sigma^\upsilon_\gamma$ is not a leaf (otherwise we would be in the first case we disposed of above), so $\theta^\upsilon(\gamma) < E$.

We extend $\gamma$ to an active $\omega$. First, take $\gamma_0 = \langle \theta^\upsilon(\gamma) + 1\rangle^\frown \gamma$. If $r_\gamma = 0$, we take $\omega = \gamma_0$. Otherwise, we repeatedly extend $\gamma_0$ to $\langle \theta^\upsilon(\gamma_0) + 1\rangle^\frown \gamma_0$, iterating this process until we get a function with sufficient domain. We take this to be $\omega$, and note that, by construction, $\omega$ is active.

For each $E' < r_\gamma$, let $\gamma_{E'} = \omega \upharpoonright (E', E)$. Then $r_{\gamma_{E'}} = E' < r_\gamma$, so $S_{<p^+_\gamma}$ is finite. Let $q' = q^\upsilon_{\rho^\upsilon_\omega(|\tau^\upsilon_\omega|-1)}$; as above, we see that $S_\perp = \bigcap_{x\in q'} S_{\perp x}$ is cofinite. If $\tau^\upsilon_\omega$ is a leaf, $(p, q', S_\perp)$ is our desired witness and we are done.

Otherwise we turn to the trial solution. Let $\hat{q} = q^\upsilon_{\hat{\rho}^\upsilon_\omega(|\hat{\tau}^\upsilon_\omega|-1)}$. Let $\hat{S}_\perp = \bigcap_{x\in\hat{q}} S_{\perp x}$. Suppose $\hat{S}_\perp$ is infinite. If $\hat{\tau}^\upsilon_\omega$ is a leaf, $(p, \hat{q}, \hat{S}_\perp)$ is the desired witness and we are done.

Suppose $\hat{\tau}^\upsilon_\omega$ is not a leaf. Let $S^*$ be the set of $y \in \hat{S}_\perp$ such that there is a $\hat{p}$ extending $p_\gamma$ with $y = \hat{p}^+$ and there exist $\sigma$ an immediate extension of $\sigma^\upsilon_\gamma$, an $e$, and $\vec{a}$ so that $L^{X\oplus\hat{p}}_\sigma((e_{\pi^\upsilon_\gamma(0)}, \dots, e_{\pi^\upsilon_\gamma}(|\sigma^\upsilon_\gamma| - 1), e), \vec{a})$ holds. If $S^*$ is infinite then $(p, \hat{q}, S^*)$ is the desired witness: if we could find an extension of $\hat{q}$ witnessing an extension $\tau$ of $\hat{\tau}^{\upsilon}_{\omega}$ in $S^*$ then we would have a witness to $\upsilon^\frown\langle(0, \omega, \tau)\rangle$, so $(p, \hat{q}, S^*)$ forces $\Theta^{X\oplus\Lambda}_{R^-;\hat{\tau}^{\upsilon}_{\omega}}(c)$ on the antichain side.

On the other hand, if $S^*$ is finite, we may choose $t$ large enough to bound $S^*$, and $(p_\gamma, q, S_\perp \upharpoonright (t, \infty))$ is the desired witness since it forces $\Theta^{X\oplus\Lambda}_{R^+;\sigma^{\upsilon}_{\gamma}}(c)$ on the chain side.

So next, consider the case where $\hat{S}_\perp$ is finite, there is an $x \in \hat{q}$ so that $S_{<x} \cap S_\perp$ is infinite. $x$ is endpoint of some chain $p^k_{\hat{\rho}^{\upsilon}_{\omega}(j)}$ for some $j < |\hat{\tau}^{\upsilon}_{\omega}|$ and some $k \leqslant k_{\hat{\rho}^{\upsilon}_{\omega}(j)}$. If there were any extension of $q'$ witnessing an extension $\tau$ of $\tau^{\upsilon}_{\omega}$ in $S_{<x} \cap S_\perp$ then we would have a witness to the node $\upsilon^\frown\langle(1, \omega, \sigma^k_{\hat{\rho}^{\upsilon}_{\omega}(j)}, \tau, j)\rangle$. $\square$

2.4. **Solving WKL.** We wish to show:

**Lemma 2.14.** *Suppose $c$ satisfies every requirement in $X$ and $U_e$ is an infinite, $\{0,1\}$-branching, $X$-computable tree. Then there is an infinite path $\Lambda$ so that $c$ satisfies every requirement in $X \oplus \Lambda$.*

We will need variants of this repeatedly, so we state and prove a mild generalization, essentially showing that the same holds if we place various restrictions on the kinds of requirements we wish to deal with.

**Lemma 2.15.** *Let $J \subseteq I \subseteq \mathbb{N}$ be given with $0 \in I \setminus J$. Suppose $c$ satisfies every requirement in $X$ with range $I$ which is transitive in every $j \in J$ and $U_e$ is an infinite, $\{0,1\}$-branching, $X$-computable tree. Then there is an infinite path $\Lambda$ so that $c$ satisfies every requirement in $X \oplus \Lambda$ with range $I$ which is transitive in every $j \in J$.*

Then Lemma 2.14 is the case with $I = \mathbb{N}$ and $J = \emptyset$.

*Proof.* It suffices to show that for any requirement $R = (T, \{K_\sigma\}, \{d_\sigma\})$, we can find an initial segment $\lambda \in U_e$ and an infinite $X$-computable $U' \subseteq U_e$ of extensions of $\lambda$ so that whenever $\Lambda$ is a branch through $U_e$, $c$ satisfies $R$ in $X \oplus \Lambda$.

We need the lexicographic ordering on $T$: if $\sigma, \tau \in T$ then $\sigma \prec \tau$ if, for the least $i$ with $\sigma(i) \neq \tau(i)$, we have $\sigma(i) < \tau(i)$.

We will describe a tree $T'$ and, for each $\upsilon \in T'$, simple block statements $K^X_\upsilon$. Each non-empty node $\upsilon$ will be associated with a $k^\upsilon \leqslant \max\{|\tau| \mid \tau \in T\}$, a $\tau^\upsilon \in T$ with $|\tau^\upsilon| = k^\upsilon$, and a monotone function $\pi^\upsilon : [1, k^\nu] \to \mathrm{dom}(\upsilon)$.

We describe the assignment of these values to $\upsilon$ inductively. Having defined $k^\upsilon, \tau^\upsilon, \pi^\upsilon$, we must describe the possible extensions of $\upsilon$. There are two types of extensions. For each $\tau' \in T$ which is an immediate extension of $\tau^\upsilon$, we have an extension $\upsilon' = \upsilon^\frown\langle(0, \tau')\rangle$ with $k^{\upsilon'} = k^\upsilon + 1$, $\tau^{\upsilon'} = \tau'$, and $\pi^{\upsilon'} = \pi^\upsilon \cup \{(k^\upsilon + 1, |\upsilon|)\}$.

Also, for each $i \in [1, k^\upsilon]$ and each $\tau' \in T$ with $|\tau'| = i$ and $\tau' \prec \tau^\upsilon$, we have an extension $\upsilon' = \upsilon^\frown\langle(1, \tau')\rangle$ with $k^{\upsilon'} = i$, $\tau^{\upsilon'} = \tau'$, and $\pi^{\upsilon'} = \pi^\upsilon \upharpoonright [1, i]$.

The first type of extension indicates that we are looking for extensions of the branches satisfying longer sequences from $T$. When this happens, it might be that different branches $\lambda$ satisfy different nodes $\sigma \in T$; in this case we have to play favorites: different branches from $T$ might have conflicting $d_\sigma$. So we choose the $\prec$-largest branch $\sigma$ such that we see a branch witnessing $\Delta^{X\oplus\lambda}_{R;\sigma}$. The second kind of extension represents backtracking on this decision: at a later stage, all branches witnessing $\Delta^{X\oplus\lambda}_{R;\sigma}$ might die out, and we have to switch to some $\sigma' \prec \sigma$. Of course, this can only happen finitely often because $U$ is finitely branching.

The auxiliary datum $b_i$ will have the form $(s_i, b'_i, r_i)$. First, consider the case where $\upsilon$ ends in $(0, \tau')$. For notational reasons, define $\pi^\upsilon(0) = -1$, $s_{-1} = 0$, and $r_{-1}(\langle\rangle) = \langle\rangle$. Then $K^X_\upsilon((b_0, \dots, b_{|\upsilon|-1}), \vec{a})$ holds if:

- $r_{|\upsilon|-1}$ assigns, to each $\lambda \in U_e$ with $|\lambda| = s_{|\upsilon|-1}$ such that $r_{\pi^\upsilon(k-1)}(\lambda \upharpoonright s_{\pi^\upsilon(k-1)}) \sqsubset \tau^\upsilon$ a sequence $r_{|\upsilon|-1}(\lambda) \in T$ with $|r_{|\upsilon|-1}(\lambda)| = k$, $r_{\pi^\upsilon(k-1)}(\lambda \upharpoonright s_{\pi^\upsilon(k-1)}) \sqsubset r_{|\upsilon|-1}(\lambda)$, and $r_{|\upsilon|-1}(\lambda) \preceq \tau^\upsilon$,
- $s_{|\upsilon|-1} \geqslant s_{|\upsilon|-2}$,
- for each $\lambda \in U_e$ with $|\lambda| = s_{|\upsilon|-1}$, there are $\vec{a}' \subseteq \vec{a}$ so that $K^{X\oplus\lambda}_{r_{|\upsilon|-1}(\lambda)}(\vec{b}', \vec{a}')$ holds.

In the other case, when $\upsilon$ ends in $(1, \tau')$, $K^X_\upsilon$ holds if:

- for each $\lambda \in U_e$ with $|\lambda| = s_{|\upsilon|-1}$, $r_{\pi^\upsilon(k)}(\lambda \upharpoonright s_{\pi^\upsilon(k)}) \prec \tau^\upsilon$, and
- $s_{|\upsilon|-1} \geqslant s_{|\upsilon|-2}$.

In either case $d'_\upsilon(\pi^\upsilon(i)) = d_{\tau^\upsilon}(i)$ and $d'_\upsilon$ constantly 0 outside the range of $\pi^\upsilon$. Clearly $\mathrm{rng}(d'_\upsilon) \subseteq I$. Below we will verify the transitivity requirement.

These requirements say that when $\Delta^X_{R';\upsilon}$ holds, each path $\lambda$ should satisfy $\Delta^{X\oplus\lambda}_{R;r(\lambda)}$ for a suitable $r(\lambda) \in T$ of length $k^\upsilon$. Note that in the second kind of block statement, $\vec{a}$ is empty—we have already arranged satisfaction of the requirement at an earlier level; the potential change is in our choice of which requirement from $T$ we are trying to satisfy.

We verify the transitivity requirement on $d'_\upsilon$. Observe how $\mathrm{rng}(\pi^\upsilon)$ changes along some path in $T'$: it is either extended by $\{|\upsilon|\}$ or reduced to some initial segment. In particular, if $\upsilon_0 \subsetneq \upsilon_1$ and $d'_{\upsilon_1}(|\upsilon_0|) = j \in J$, so $d'_{\upsilon_1}(|\upsilon_0|) \neq 0$, then we have $|\upsilon_0| = \pi^{\upsilon_1}(|\tau^{\upsilon_0}|)$, and therefore for every $\upsilon$ with $\upsilon_0 \sqsubseteq \upsilon \sqsubseteq \upsilon_1$, we have $\pi^\upsilon \upharpoonright |\tau^{\upsilon_0}| = \pi^{\upsilon_0} \upharpoonright |\tau^{\upsilon_0}|$. In particular, if $d'_{\upsilon_0}(i) = d'_{\upsilon_1}(|\upsilon_0|)$ then we have $i = \pi^{\upsilon_0}(i') = \pi^{\upsilon_1}(i')$, $d_{\tau^{\upsilon_0}}(i') = j$, $d_{\tau^{\upsilon_1}}(|\tau^{\upsilon_0}|) = j$, and so by the transitivity of $j$ in $T$, $d_{\tau^{\upsilon_1}}(i') = j$, so $d'_{\upsilon_1}(i) = j$.

We must check that satisfaction of our requirement ensures that we can choose a $\lambda$ forcing satisfaction of the original requirement. Suppose we satisfy $\Theta^X_{R;\upsilon}(c)$. There are finitely many leaves $\lambda \in U_e$ with $|\lambda| = s_{|\upsilon|-1}$ and $r_{\pi^\upsilon(k^\upsilon)}(\lambda) = \tau^\upsilon$. (There must be some such leaves, since otherwise we would also satisfy an extension of $\upsilon$.) Consider the tree $U'' \subseteq U_e$ consisting of those $\lambda'$ extending some such $\lambda$ but not finding any witnesses to any extension of $\tau^\upsilon$. If $U''$ were finite, we would again satisfy an extension of $\upsilon$, so $U''$ must be infinite. Some $\lambda$ must have infinitely many extensions in $U''$; by choosing this $\lambda$ and $U' \subseteq U''$ consisting of extensions of $\lambda$, we have forced $\Theta^{X\oplus\Lambda}_{R;\tau^{\upsilon}}(c)$. □

2.5. **Constructing STS(2).**

**Lemma 2.16.** *There is a computable stable $c : [\mathbb{N}]^2 \to \mathbb{N}$ satisfying all requirements in $\varnothing$.*

Again, we prove a more general version that will include later cases.

**Lemma 2.17.** *Let $J \subseteq I \subseteq \mathbb{N}$ with $0 \in I\backslash J$. There is a computable stable $c : [\mathbb{N}]^2 \to I$ transitive in every color in $J$ and satisfying all requirements in $\varnothing$ with range $I$ which are transitive in every color in $J$.*

Again, Lemma 2.16 is the case with $J = \varnothing$ and $I = \mathbb{N}$.

*Proof.* This is a standard finite injury priority argument. Informally, we place all requirements with range $I$ transitive in every color in $J$ in order, and every time we find witnesses violating a negative requirement component, we remember the witnesses, restrain them so future colors comply with the corresponding positive requirement component, and injure all lower priority requirements; that requirement is then witnessed along a longer branch $\upsilon$. Since each requirement has a finite tree, each requirement eventually stops acting, either because some negative requirement component holds or because we reach a leaf.

More formally, we proceed as follows. We order the requirements $R_0, R_1, \ldots$. At each stage $s$ we have fixed:

- $c_s : [s]^2 \to I$ transitive in each color in $J$,
- for $r < s$, $\upsilon_{s,r} \in T_r$, $b_{s,r,0}, \ldots, b_{s,r,|\upsilon_{s,r}|-1}$, $\vec{a}_{s,r,0}, \ldots, \vec{a}_{s,r,|\upsilon_{s,r}|-1}$, $t_{s,r}$, and sets $A_{s,r,j}$ so that:
  - for each $i < |\upsilon_{s,r}|$, $K_{\upsilon_{s,r}\upharpoonright(k+1)}((b_{s,r,0}, \ldots, b_{r,i}), \vec{a}_{s,r,i})$,
  - if $j \neq j'$ then $A_{s,r,j} \cap A_{s,r',j'} = \varnothing$,
  - each $\vec{a}_{s,r,i} \in A_{s,r,d_{\upsilon_{s,r}}(i)}$,
  - if $r' < r$ then $t_{s,r'} \leqslant t_{s,r}$ and $t_{s,r'} < \vec{a}_{s,r,i}$,
  - if $b \in A_{s,r,i}$, $i \in J$, $a < b$, and $c(a,b) = i$ then $a \in A_{s,r,i}$.

We will have $c_s \subseteq c_{s+1}$. The sets $\bigcup_{r\leqslant s} A_{s,r,i}$ are approximations to $A^*_i(c)$. If $a \notin \bigcup_{r\leqslant s}\bigcup_i A_{s,r,i}$, we will treat $a$ as if it belongs to some $A_{s,r,0}$.

Suppose we have constructed up to stage $s$. Define $c_{s+1}(n, s+1)$ for $n < s+1$ by setting $c_{s+1}(n, s+1) = i$ if $n \in A_{s,r,i}$ for some $r$. (The closure condition on $A_{s,r,i}$ ensures transitivity of $c$.) Let $r < s$ be least (if there is any) so that there is some $b$, some $\vec{a} \in (t_{s,r}, s+1)$, and some $\upsilon$ an immediate extension of $\upsilon_{s,r}$ in $T_r$ so that $K_\upsilon((b_{s,r,0}, \ldots, b_{s,r,|\upsilon_{s,r}|-1}, b), \vec{a})$ holds; otherwise $r = s$. For $r' < r$, we have $\upsilon_{s+1,r'} = \upsilon_{s,r'}$, $b_{s+1,r',i} = b_{s,r',i}$, $\vec{a}_{s+1,r',i} = \vec{a}_{s,r',i}$, $t_{s+1,r'} = t_{s,r'}$, and $A_{s+1,r',i} = A_{s,r,i}$.

If $r < s$, let $\upsilon_{s+1,r} = \upsilon$, $b_{s+1,r,|\upsilon|-1} = b$, $b_{s+1,r,i} = b_{s,r,i}$, $\vec{a}_{s+1,r,|\upsilon|-1} = \vec{a}$, $\vec{a}_{s+1,r,i} = \vec{a}_{s,r,i}$, and $t_{s+1,r} = s+1$. Take $A_{s+1,r,j}$ to consist of those $\vec{a}_{s+1,r,i}$ with $d_\upsilon(i) = j$, together with any elements required by the closure condition.

Note that if $a \in A_{s+1,r',j}$ for some $r' < r$ then $c_s(a,b) = j$ for any $b > t_{s,r'}$, so in particular any $\vec{a}_{s+1,r,i}$, so if $b \in A_{s+1,r,j}$, there is no conflict with having $a \in A_{s+1,r,j}$ as well.

For $r' \in (r,s]$ (or $r' = s$ if $r = s$), set $\upsilon_{s+1,r'} = \langle\rangle$, $A_{s+1,r',i} = \emptyset$, $t_{s+1,r'} = s+1$, and $A_{s+1,r',j} = \emptyset$.

We only injure a requirement $R_j$ if we make the node $\upsilon_{s,j'}$ longer for some $j' < j$, so a requirement is injured only finitely many times. In particular, there is a limiting node $\upsilon_j = \lim_s \upsilon_{s,j}$. The witnesses $b_{s,j,0}, \dots, b_{s,j,|\upsilon_j|-1}$ and $\vec{a}_{s,j,0}, \dots, \vec{a}_{s,j,|\upsilon_j|-1}$ also stabilize to witnesses $b_{j,0}, \dots, b_{j,|\upsilon_j|-1}$ and $\vec{a}_{j,0}, \dots, \vec{a}_{j,|\upsilon_j|-1}$. In particular, these witness $\Delta_{R_j;\upsilon_j}(c)$. Furthermore, if $\upsilon_j$ is not a leaf, $t_{s,j}$ stabilizes to some $t_j$ larger than any witness to any lower priority requirement, and there do not exist $b, \vec{a}$ and $\upsilon$ extending $\upsilon_j$ with $\vec{a} > t_j$ so that $K_{R_j;\upsilon}((b_0, \dots, b_{|\upsilon|-1}), \vec{a})$, since if there were, we would have taken $\upsilon_{s,j} = \upsilon$ at some stage, so $\Theta_{R_j;\upsilon_j}(c)$ holds.

Finally, we check that $c$ is stable; it suffices to show that for each $n$, there is some $s, i$ such that for all $s' \geqslant s$, $n \in A_{s,i}$. But $n$ can only be moved from one $A_i$ to another when some requirement $\leqslant n$ acts, which only happens finitely many times. □

## 3. Separating **SProdWQO**

3.1. **Separating from ADS.** In this section we construct a computable instance $c$ of **SProdWQO** (and, a fortiori, of **SCAC**) and a Turing ideal $\mathcal{I}$ which has no solution to $c$, but does satisfy both $\mathbf{trRT^2_k}$ for all $k$ and **WKL**.

**Definition 3.1.** An **SProdWQO**-*requirement* is a requirement $R = (T, \{K_\alpha\}_{\sigma \in T}, \{d_\sigma\}_{\sigma \in T})$ with range $\{0,1,2\}$ transitive in both colors 1 and 2.

Lemmata 2.8, 2.15 and 2.17 apply with $J = \{1,2\}$, $I = \{0,1,2\}$, so we have:

**Lemma 3.2.** *If $c$ satisfies all* **SProdWQO**-*requirements in $X$ then whenever $B$ is an $X$-computable infinite set, there exist $a,b,c,d \in B$ with $c(a,b) = 1$ and $c(c,d) = 2$.*

**Lemma 3.3.** *If $c$ satisfies all* **SProdWQO**-*requirements in $X$ and $U$ is an infinite $X$-computable $\{0,1\}$-branching tree then there is an infinite branch $\Lambda$ so that $c$ satisfies all* **SProdWQO**-*requirements in $X \oplus \Lambda$.*

**Lemma 3.4.** *There is a computable stable $c : [\mathbb{N}]^2 \to \{0,1,2\}$ transitive in the colors 1 and 2 satisfying every* **SProdWQO**-*requirement in $\emptyset$.*

We first give our argument showing that we can satisfy **ADS**.

**Lemma 3.5.** *Suppose $c$ satisfies every* **SProdWQO**-*requirement in $X$ and $\prec$ is a linear ordering. Then there is an infinite $\prec$-monotone sequence $\Lambda$ so that $c$ satisfies every* **SProdWQO**-*requirement in $X \oplus \Lambda$.*

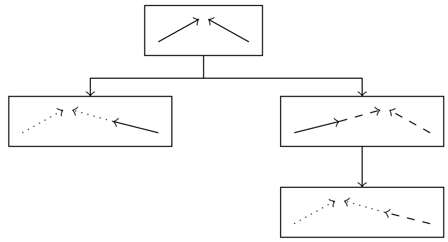

Figure 5.

While the bookkeeping in this argument is quite unwieldy, the general idea is perhaps less complicated than that of Lemma 2.13. We describe the beginnings of the induction that make up the core of the proof.

When $T^+$ and $T^-$ each consist of a single non-empty node, we can proceed as above: we wait for a suitable split pair $(p, q)$. For us a split pair is a pair $(p, q)$ where $p$ is an increasing sequence, $q$ is a decreasing sequence, and they share the endpoint $p^+ = q^+$. If we never find such a split pair, either $\Theta^{X\oplus p}_{R^+;\langle 0\rangle}(c)$ holds for all suitable $p$ or $\Theta^{X\oplus q}_{R^-;\langle 0\rangle}(c)$ holds for all suitable $q$. If we find a split pair, one of $p$ or $q$ must be a valid extension to our construction.

This is the basis of what we will call a *process of type* $\{(1,1)\}$: we are able to produce a split pair $(p, q)$ where each part witnesses a node of length 1 from the corresponding tree.

More generally, by a process with a type $K' \subseteq K = [1, D] \times [1, E]$ (where $D = \max\{|\sigma| \mid \sigma \in T^+\}$ and $E = \max\{|\tau| \mid \tau \in T^-\}$), we mean a description of a tree where each leaf is witnessed by a split pair $(p, q)$ where, for some $(d, e) \in K'$, $p$ witnesses a node of length $d$ and $q$ witnesses a node of length $e$. The key argument will be showing that, given a two processes, one of type $\{(1, e)\}$ and one of type $\{(1, e + 1), (d, e)\}$, we can produce a process of type $\{(1, e + 1), (d + 1, e)\}$.

Suppose $T^+ = T^- = \{\langle\rangle, \langle 0\rangle, \langle 0, 0\rangle\}$—that is, a single branch of length 2. In Figure 5, we show a process of type $\{(1, 2)\}$ (so, a fortiori, a process of type $\{(1, 3), (1, 2)\}$), which is not so complicated. Each solid box indicates a node in our tree. In Figure 6, we show the combination with $e = 2$ and $d = 1$, which gives a process of type $\{(1, 3), (2, 2)\}$; since $(1, 3)$ is impossible in this case, this is simply a process of type $(2, 2)$.

Figure 6 requires some more explanation. We use doubled lines to indicate that the branches are active at this node; when a branch is drawn with a single line, it is inactive in a strong sense: we specifically require that $d_\sigma(i) = 0$. The overarching structure is the same as the $\{(1, 2)\}$ process from Figure 5, with some additional steps added in. We use dotted lines to indicate that we are running the side process (which, in this case, happens be the same process as Figure 5, but in general might be different from the overarching structure) to produce a pair of type $(1, 2)$; this subprocesses correspond to inserting an entire copy of the tree from Figure 5 in that location, and we only display the outcome at the leaves of that subproecess. After succeeding

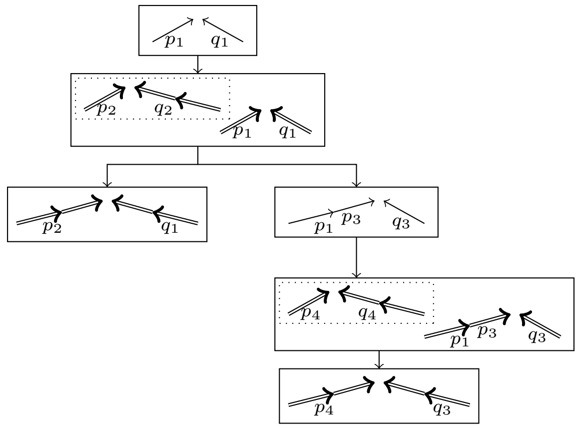


FIGURE 6.

at the first stage, leading to finding the pair $(p_1, q_1)$, we deactivate these nodes and run the side process. If the side process terminates with its own pair $(p_2, q_2)$, we now activate both this pair and $(p_1, q_1)$. Here we modify the original process: the original process looked for a pair $(p, q)$ where $p$ is a fresh increasing sequence and $q$ is an extension of $q_1$. Instead, we now look for a pair $(p, q)$ where $p$ extends $p_2$.

In the other branch, we find the pair $(p_3, q_1)$. Again, we then deactivate that pair and run the side process again. (The pair $(p_2, q_2)$ is no longer useful, and we discard it; we cannot keep it because there may be transitivity commitments between the witnesses to $(p_2, q_2)$ and the witnesses to $(p_3, q_3)$ which we might not be able to respect at later stages if we kept both.) If the new side process finds a pair $(p_4, q_4)$ of type $(1, 2)$, we now activate both parts and run the last step of the main process, except that instead of looking for a new increasing sequence of length 1, we look for an extension of $p_4$ to a sequence of length 2.

The pattern is that every time the next step of the overarching process *might* have produced a $(1, 2)$ pair (and then finished), we pause, run the side process, and then resume looking for a $(2, 2)$ pair instead. Our ability to deactivate the main process while the side process runs is crucial to being able to maintain the transitivity requirements.

Note that there is no requirement on the relationship between the endpoints of the two parts. For example, consider the second to last node, where we hope to extend $p_4$ and $q_1$. It might be that $q_1^+ \prec p_4^+$, which would make this impossible. But in this case, either there infinitely many points $x$ with $x \prec q_3^+ = p_4^+$, in which case $q_3$ is a valid extension, or there are infinitely many points $x$ with $p_3^+ = q_1^+ \prec p_4^+ \prec x$, in which case $p_3$ is a valid extension.

The last point we need to discuss is how the pattern of inactivations above prevents the transitivity requirement from interefering with when we need to activate sequences. It will be convenient later to have some terms for discussing this. Let $p$ be some witnessing chain at some node $\upsilon$, witnessing some $\sigma \in T^+$, so we have $\pi : \mathrm{dom}(\sigma) \to \mathrm{dom}(\upsilon)$ indicating the stages at which $p$ was constructed; recall that $p$ is active when $d_\upsilon(\pi(i)) = d^+_\sigma(i)$ for all $i \in \mathrm{dom}(\sigma)$. Let us say the *$i$-th stage of $p$ is restrained to $b$* if $d_\upsilon(\pi(i)) = b$. We may similar definitions for a witnessing chain $q$.

The problem would be this: at some node $\upsilon$, we want two chains, say $p$ and $q$ (though they could also be two chains on the same side, say, $p_4$ and $p_3$ as in Figure 6), to be active, and we need $d_\upsilon(\pi(i)) = 0$ while $d_\upsilon(\rho(j)) = 1$, but we had $d_{\upsilon\restriction j}(\pi(i)) = 1$—that is, at the stage $\rho(j)$ where the $j$-th component of $q$ was constructed, the $i$-th stage of $p$ was restrained to 1. In our construction, this would mean that both $p$ and $q$ were active at that stage (because if $p$ were inactive, all its stages would be restrained to 0, and $q$ must have been extended, and only active chains can be extended). We have prevented the problem in the simplest possible way: if two witnessing chains are only ever active at the same stage once, and one of those chains will be discarded forever after that stage

*Proof.* The proof is similar to the proof of Lemma 2.11. Again, it suffices to assume that $\prec$ is stable-ish as witnessed by $V$, and we again force with conditions $(p, q)$ where $p^+ \in V$, $q^+ \notin V$. Again, it suffices to show:

> $(*)$ Suppose $R^+$ and $R^-$ are requirements and $(p, q)$ is a condition. Then there is a condition $(p', q')$ extending $(p, q)$ which either forces $R^+$ on the increasing side or $R^-$ on the decreasing side.

As in Lemma 2.11, we can assume that $d^+_\sigma(|\sigma| - 1) = 0$ for any $\sigma \in T^+$, and a similar assumption for $T^-$. Recall that a split pair is a pair $(p', q')$ with $p \sqsubseteq p'$, $q \sqsubseteq q'$, and $(p')^+ = (q')^+$. Let $D = \max\{|\sigma| \mid \sigma \in T^+\}$ and $E = \max\{|\tau| \mid \tau \in T^-\}$. Let $K$ be the set of pairs $(d, e)$ with $d \in [1, D]$ and $e \in [1, E]$.

We build up a tree of nodes for our requirement using "processes", which are sub-trees whose leaves promise us split pairs $(p', q')$ witnessing nodes from $T^+$ and $T^-$, but not necessarily leaves.

When $K' \subseteq K$, a *process of type $K'$* is a requirement $R'$ so that:

- for each non-leaf node $\upsilon$, $\Theta^X_{R';\upsilon}(c)$ implies that either there is a $p'$ forcing $R^+$ on the chain side or a $q'$ forcing $R^-$ on the antichain side, and
- for each leaf $\upsilon$, there is a pair $(d, e) \in K'$, a $\sigma \in T^+$ with $|\sigma| = d$, and a $\tau \in T^-$ with $|\tau| = e$ so that $\Delta^X_{R';\upsilon}(c, \cdots)$ implies the existence of a split pair $(p', q')$ and suitable witnesses so that $\Delta^{X\oplus p'}_{R^+;\sigma}(c, \cdots)$ and $\Delta^{X\oplus q'}_{R^-;\tau}(c, \cdots)$.

That is, if we place a copy of $R'$ as some sub-tree of our actual requirement $R$, the only case in which we reach the end-nodes are when we have found a split pair $(p', q')$.

It is easy to describe a process of type $\{(1,1)\}$: it is a single node which for a split pair $(p', q')$, $b_0, d_0, \vec{a}_0, \vec{c}_0$, so that, for some $\sigma, \tau$ with $|\sigma| = |\tau|$, both $\Delta^{X\oplus p'}_{R^+;\sigma}(c, b_0, \vec{a}_0)$ and $\Delta^{X\oplus q'}_{R^-;\tau}(c, d_0, \vec{c}_0)$ hold. If the program terminates, we have constructed a split pair of type $(1,1)$. If the program runs forever, we must satisfy either $\Theta^{X\oplus p}_{R^+;\Diamond}(c)$ or $\Theta^{X\oplus q}_{R^-;\Diamond}(c)$.

For notational convenience, we will refer to processes for sets $K'$ which may include $(1, E+1)$ or $(D+1, e)$, even though $(1, E+1), (D+1, e) \notin K$; such a process is equivalent to one for $K' \cap K$. (That is, we say "the process can terminate after constructing a split pair of type $(1, E+1)$", but this case can never occur, because there are no such split pairs, so we ignore that case.)

Suppose we have a process $R'_d$ of type $K'_d = \{(1, e+1), (d, e)\}$ and a process $R'_1$ of type $K'_1 = \{(1, e)\}$. We describe a process of type $\{(1, e+1), (d+1, e)\}$.

We modify the process $R'_1$ as follows. Consider any node such that one of its children is a leaf. In the modified process, we deactivate all chains at this node, and then insert a copy of $R'_d$ below this node. Each leaf of this copy of $R'_d$ which finishes with a pair of type $(1, e+1)$ is also a leaf of the modified process. Otherwise, the leaf from $R'_d$ becomes a node of our larger process which promises a pair $(p', q')$ witnessing some $(\sigma', \tau')$ with $|\sigma'| = d$ and $|\tau'| = e$.

We now consider the children of this node in our new process. First, for each immediate extension of $\tau'$, we have a child node corresponding to the case where we find a split pair $(p^*, q'')$ of type $(1, e+1)$ with $q''$ extending $q'$. Next, for each child of the original node from $R'_1$, we have a child node. If the node from $R'_1$ was a leaf, and therefore would have produced a pair $(p^*, q^*)$ of type $(1, e)$, this becomes a modified leaf producing a pair $(p'', q^{**})$ of type $(d+1, e)$ where $p''$ extends $p'$. All other child nodes from $R'_1$ become children in our new process without modification.

In particular, the only leaves of the new process are leaves from one of the copies of $R'_d$ of type $(1, e+1)$, the leaves we added of type $(1, e+1)$, or modified leaves from $R'_1$ which are of type $(d+1, e)$.

Iteration of this method gives the desired process. We have a process of type $\{(1,1)\}$. Given a process of type $\{(1, e)\}$, we apply this combination to obtain a process of type $\{(1, e+1), (2, e)\}$, and by repeating $\{(1, e+1), (d, e)\}$ for any $d$. In particular, we get a process of type $\{(1, e+1), (D+1, e)\}$, which is the same as a process of type $\{(1, e+1)\}$. Inductively, we have processes of type $\{(1, e)\}$ for all $e$. In particular, applying the first iteration again, we have processes of type $\{(1, E+1), (d, E)\}$ for each $d$, which is the same as a process of type $\{(d, E)\}$. Finally, we obtain a process of type $\{(D, E)\}$, which suffices to give the desired extensions. $\square$

3.2. **Separating from $\mathbf{trRT^2_k}$.** We need to generalize the ideas of the previous subsection to $\mathbf{trRT^2_k}$. The general ideas are the same, but the bookkeeping is slightly more complicated because we now have $k$ different processes we need to interleave.

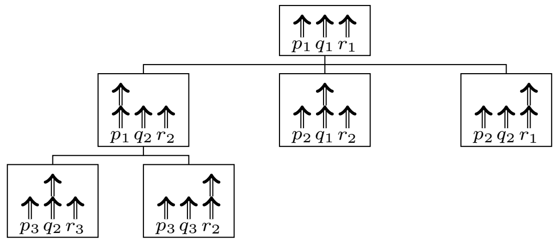

FIGURE 7. Process of type $\{(1,2,1),(1,1,2)\}$

For each $i$ we will have a requirement $R_i$ with a tree $T_i$ so that $\max\{|\sigma| \mid \sigma \in T_i\} = D_i$. We will work with split $k$-tuples $(p_1,\dots,p_k)$ where $p_i$ is a chain in the $i$-th color and $p_1^+ = \cdots = p_k^+$ (so that there is some $i$ so that, for infinitely many $x$, $c(p_i^+, x) = i$).

We take $K = \prod_i [1, D_i]$, and construct processes of type $K' \subseteq K$, by which we mean requirements with the property that for each each leaf there is $(d_1,\dots,d_k) \in K'$ so that the leaf promises the existence of a split pair $(p_1,\dots,p_k)$ and, for each $i$, a $\sigma_i \in T_i$ with $|\sigma_i| = d_i$ so that $\Delta^{X\oplus p_i}_{T_i;\sigma_i}(c,\cdots)$ holds.

We want to work towards processes of "larger" type. It is clear that, say, finding a split pair of type $(1,2,2)$ represents more progress than a pair of type $(1,2,1)$; we work lexicographically, so we also consider a pair $(1,1,2)$ to be further progress than a pair of type $(1,3,1)$. (This is consistent with what we did above, where we considered a slightly longer antichain to be more progress than a much longer chain.)

We illustrate the $k = 3$, $D_1 = D_3 = 2$ and $D_2 = 3$ case concretely. Techniques from above give us a process of type $\{(1,2,1),(1,2,2)\}$ fairly easily, as illustrated in Figure 7. Again we double lines to indicate that the sequence is active (which is always the case in this diagram). We no longer indicate the arrows pointing at each other, so one should remember that each triple of arrows share an endpoint, and in particular, one of them must be extensible. When a sequence is inactive, all witnesses from that block are temporarily being colored 0 with all new elements, allowing us to find new witnesses without creating transitivity obligations between them.

In Figure 8 we describe a process of type $\{(2,2,1),(1,3,1),(1,1,2)\}$; we do this by using the process from Figure 7 as our template, and adding in additional copies of the same process as intermediate steps. Again the dotted boxes indicate that we have inserted a copy of the sub-process, and are only displaying the conclusion of that sub-process. Furthermore, each subprocess might terminate in a leaf giving witnesses of type $(1,1,2)$, which we can take as a leaf of our larger tree.

Again, our principle is that we modify Figure 7 by looking at nodes which have a child which is a leaf of type $(1,2,1)$; at each such leaf, we add a new copy of the subprocess. For example, consider the third level of the tree

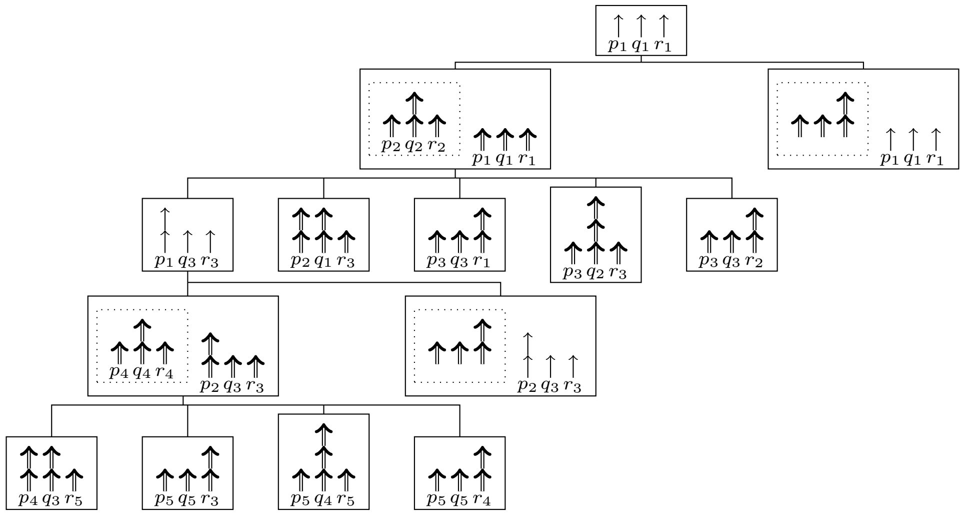

FIGURE 8. Process of type $\{(2,2,1),(1,3,1),(1,1,2)\}$

in Figure 8, consisting of all possible outcomes following the left node on the second level. There are two split triples, so in principle nine different pairs that might be extended (that is, perhaps $p_1$ and $p_2$ are extended, or $p_1$ and $q_2$, etc.). If $r_2$ or $q_2$ gets extended, we get one of the left two nodes on the third row, both of which are witnesses of the desired type. Similarly, if $r_1$ is extended, we can get a leaf with a witness of type $(1,1,2)$ and ignore what happens in the other triple. (If some combination happens, say $r_1$ and $q_2$ could both be extended, which leaf we end up at depends on which witnesses get found first.) If $q_1$ is extended, this is the case where we need our modification: instead of producing a leaf with a witness of type $(1,1,2)$, we hope to extend $p_2$, thereby getting a leaf of type $(2,2,1)$ (we need to make sure that if $p_2$ is not extended, we end up in some other desirable case, as we do here). Finally, if $p_1$ is extended, this leads to a non-leaf in the original tree, so we let it bring us to a non-leaf here, discarding the second split triple as no longer useful.

For completeness, we include a process of type $\{(1,3,1),(1,1,2)\}$ in Figure 9. The only difference is that now the subprocess is of type $\{(2,2,1),(1,1,2)\}$; this has the effect of ruling out some nodes because, $p_2$ and $p_4$ now cannot be extended (we leave the missing nodes as empty spots, not only because it means we don't have to modify that source code as much, but also to make the comparison easier). (If instead we had $D_1 > 2$, these nodes would exist with witnesses of type $(3,2,1)$, and we would get a process of type $\{(3,2,1),(1,3,1),(1,1,2)\}$ in this way.)

We produce a process of type $\{(2,3,1),(1,1,2)\}$ similarly: this time we begin with a process of type $\{(1,3,1),(1,1,2)\}$, and use the same process as a subprocess. Another iteration, taking a process of $\{(1,3,1),(1,1,2)\}$ and using a subprocess of type $\{(2,3,1),(1,1,2)\}$ gives us a process of type

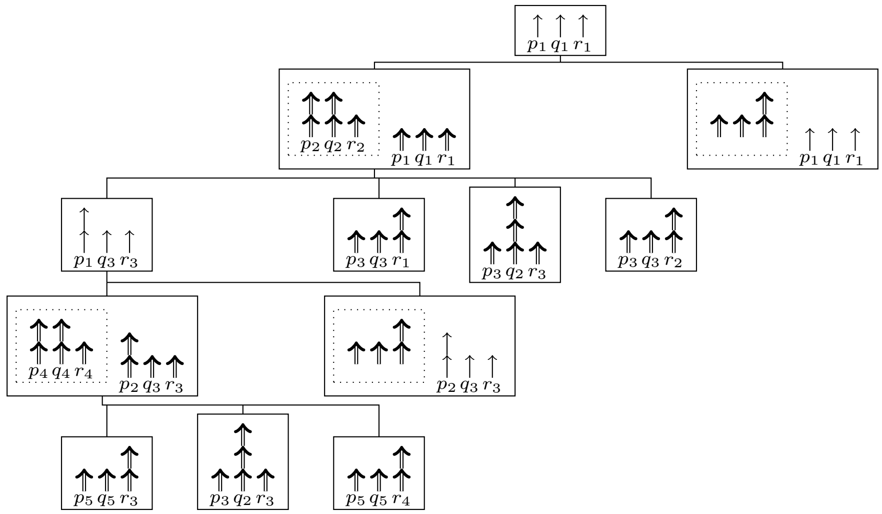

FIGURE 9. Process of type $\{(1,3,1),(1,1,2)\}$

$\{(1,1,2)\}$. We can do another round of iterations, obtaining a process of type $\{(2,1,2),(1,2,2)\}$ and then a process of type $\{(1,2,2)\}$.

We need a new idea, however, to complete this to a process of type $\{(2,2,2),(1,3,2)\}$. Consider the thing we would like to do: begin with a process of type $\{(1,2,2)\}$ and use the same process as the subprocess. The difficulty is that the subprocess might extend the $q$ component, giving us a witness of type $(1,3,1)$; but this is useless to us: we need witnesses of type $(1,3,2)$ instead.

The difficulty here is that, to use our idea of inserting subprocesses, we take a node where the outer process intended to create a witness of length 1, and instead have it produce a witness of length $k+1$ (where $k$ was the length of the witness produced by the subprocess). But at the relevant node in the process of type $\{(1,2,2)\}$, it is already has an existing $q$ component it might extend, so it can't try to extend the $q$ component coming from the subprocess.

Our subprocess will give us a node with a witness $(p,q,r)$ of type $(1,2,2)$; either the $p$ or $q$ is extensible (if there are cofinitely many $x$ with $c(r^+,x)=3$ then $r$ is the extension we need, and we will never move to another node). If $q$ is extensible, we wish we were about to produce a leaf with witnesses of type $(1,1,2)$, since we could then ask to extend $q$. On the other hand, if $p$ is extensible, we want to be producing a leaf of type $(1,2,2)$.

Since we can't predict which of $p$ or $q$ will be extensible, we need two different "outer" processes, so that we can extend either. Here, unfortunately, the iteration process becomes a bit complicated. For simplicity, we imagine that the processes of type $\{(1,1,2)\}$ and $\{(1,2,2)\}$ are given by the simplified trees in Figure 10; we treat the intermediate stages as a black box, and only look at the leaves, since we will need to modify the leaves in our combined

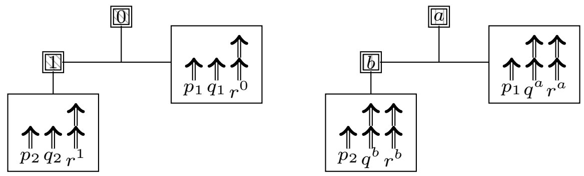

FIGURE 10. Abstracted processes of type $\{(1,1,2)\}$ and $\{(1,2,2)\}$

process. The doubled lines around the box indicate that the witnesses to the node are active (and therefore eligible to be extended in the next node).

In Figure 11 we describe a new subprocess whose goal which either terminates with witnesses of type $(2,2,2)$ or of type $(1,3,2)$, *or* which makes it to stage $b$ (the second node) of our abstract process of type $\{(1,2,2)\}$. This process is modeled on the process of type $\{(1,1,2)\}$, except that each time we are at a node whose child is a leaf, we make a modification. For example, consider the first such node, the one labeled 0; we see in Figure 11 that after reaching node 0, we deactivate all witnesses and begin a new process of type $\{(1,2,2)\}$, allowing it get to stage $a$, and then deactivate that as well. We then insert a subprocess of type $\{(1,2,2)\}$. One we have all three, we allow them all to run simultaneously; the possible outcomes are either that our main process advances to stage 1, or our process of type $\{(1,2,2)\}$ advances to stage $b$, or $q_1$ is extensible (which, together with the main process producing a witness of type $(1,1,2)$ gives us the desired witness of type $(1,3,1)$), or $p_1$ is extensible (which, together with the other subprocess producing a witness of type $(1,2,2)$ gives us the desired witness of type $(2,2,2)$).

To construct an actual process of type $\{(2,2,2),(1,3,2)\}$, we repeat the same idea, but now using the subprocess we just created, which either gives us the desired witnesses or gives us the witnesses to stage $b$ from the process of type $\{(1,2,2)\}$.

**Lemma 3.6.** *Suppose $c$ satisfies every* **SProdWQO***-requirement in $X$ and $c^* : [\mathbb{N}]^2 \to [1,k]$ with all colors transitive. Then there is an infinite $c^*$-homogeneous set $S$ so that $c$ satisfies every* **SProdWQO***-requirement in $X \oplus \Lambda$.*

*Proof.* Our conditions are tuples $(p_1,\dots,p_k)$ where each $p_i$ is homogeneously colored $i$ and there are infinitely many $x$ so that, for each $a \in p_i$, $c^*(a,x) = i$. Given requirements $R_1,\dots,R_k$, we must find a condition $(p'_1,\dots,p'_k)$ with each $p_i \sqsubseteq p'_i$ so that some $R_i$ is forced.

A *split $k$-tuple* is a tuple $(q_1,\dots,q_k)$ with each $p_i \sqsubseteq q_i$ and $(q_1)^+ = \cdots = (q_k)^+$; it follows that there is at least one $i_0$ so that, taking $p'_{i_0} = q_{i_0}$ and $p'_i = p_i$ for $i \neq i_0$, $(p'_1,\dots,p'_k)$ is a condition.

For each $i \leqslant k$, $D_i = \max\{|\sigma| \mid \sigma \in T_i\}$, and we take $K = \prod_i [1,D_i]$. The notion of constructing a split pair of type $(d_1,\dots,d_k) \in K$ and a process of type $K' \subseteq K$ generalize immediately. Again, we can immediately describe a process of type $\{(1,\dots,1)\}$—we simply wait to find any split $k$-tuple

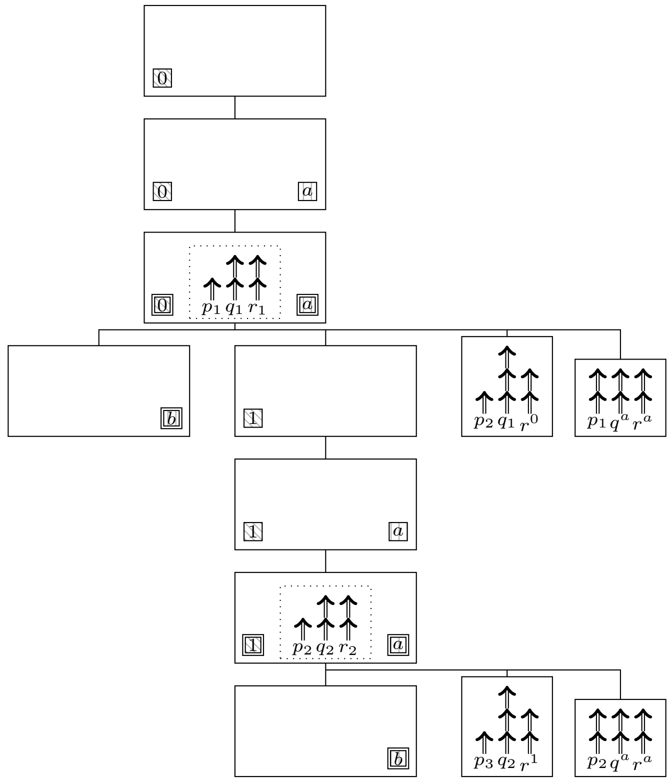


FIGURE 11. Subprocess for process of type $\{(2,2,2),(1,3,2)\}$

$(q_1,\dots,q_k)$ with each $q_i$ witnessing a $\sigma_i \in T_i$ of length 1, with failure to find such a $k$-tuple ensuring that some $\Theta^{X\oplus p_i}_{R_i;\diamondsuit}$ will be satisfied.

We place tuples $\vec{d}$ in reverse lexicographic order, so $\vec{d} < \vec{d'}$ if there is an $i$ so that $d_j = d'_j$ for $i < j$, and $d_i < d'_i$. $(1,\dots,1)$ is the smallest element in this ordering. Given some $\vec{d} \in (d_1,\dots,d_k)$, we define $\vec{d}^{+i} = (1,\dots,1,d_i+1,d_{i+1},\dots,d_k)$—that is,

$$d^{+i}_j = \begin{cases} 1 & \text{if } j < i \\ d_j+1 & \text{if } j = i \\ d_j & \text{if } j > i \end{cases}.$$

We define

$$K_{\vec{d}} = K \cap (\{\vec{d}\} \cup \{\vec{d}^{+i} \mid \exists j < i\ d_j \neq 1\}).$$

So $K_{(1,\dots,1)} = \{(1,\dots,1)\}$ while

$$K_{(1,2,1,2,1)} = \{(1,2,1,2,1),(1,1,2,2,1),(1,1,1,3,1),(1,1,1,1,2)\}.$$

We will show by induction on $\vec{d}$ that we can construct a process of type $K_{\vec{d}}$.

As noted in the discussion above, having a process of type $K_{\vec{d}}$ is not, by itself, enough to create a process of type $K_{\vec{d'}}$ (where $\vec{d'}$ is the successor of

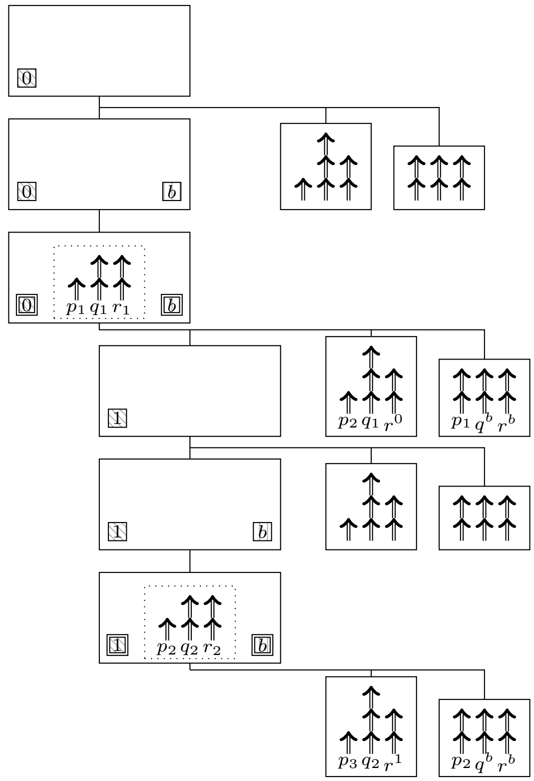


FIGURE 12. Process of type $\{(2,2,2),(1,3,2)\}$

$\vec{d}$): we may need processes the form $K_{\vec{e}}$ for various $\vec{e} \leqslant \vec{d}$ as well. We now identify precisely which values we need.

We define $\vec{d}^{\restriction i}$ by

$$d_j^{\restriction i} = \begin{cases} 1 & \text{if } j < i \\ d_j & \text{if } i \leqslant j \end{cases} .$$

Let $I(\vec{d}) \subseteq [1,k]$ be the set of $i$ so that $d_i \neq 1$ and let $Z(\vec{d}) = \{\vec{d} \restriction i \mid i \in I\}$. So $Z(1,2,1,2,1) = \{(1,2,1,2,1),(1,1,1,2,1)\}$.

Finally, we define $K_{\vec{d}}^{+} = K \cap \{\vec{d}^{+i} \mid i \leqslant k\}$.

**Claim.** Suppose that for each $\vec{e} \in Z(\vec{d})$ there is a process of type $K_{\vec{e}}$. Then there is a process of type $K' = K_{\vec{d}}^{+} \cup \bigcup_{\vec{e} \in Z(\vec{d})} (K_{\vec{e}} \backslash \{\vec{e}\})$.

*Proof.* When $\vec{d} = (1,\dots,1)$, so $I(\vec{d}) = \varnothing$ and therefore we need a process of type $K_{\vec{d}}^{+} = \{(2,1,\dots,1),(1,2,1,\dots,1),\dots,(1,\dots,1,2)\}$, we need a tree of height 2: the first node finds a split pair of type $(1,\dots,1)$, and the second node waits for any extension to a split pair of type $(1,\dots,1,2,1,\dots,1)$.

So we may assume $\vec{d} \neq (1,\dots,1)$, so in particular $Z(\vec{d})$ is non-empty. Let $Z(\vec{d}) = \{\vec{e}_1,\dots,\vec{e}_m\}$, and assume $\vec{e}_m = \vec{d}$. For each $j \leqslant m$ we have a process of type $K_{\vec{e}_j}$ given by a tree in which all nodes have height at most $c_j$.

Suppose we are at some node in our large process, and that at previous levels we have constructed, for each $j < m$, witnesses to a node $\tau_j$ from the process of type $K_{\vec{e}_j}$. We proceed as follows; first, we deactivate all other witnesses and insert a copy of the process of type $K_{\vec{d}}$. Each leaf of this subprocess either gives witnesses of type $\vec{d}^{+i} \in K'$, and is therefore a leaf of our larger process, or gives witnesses of type $\vec{d}$. Above a node with witnesses of type $\vec{d}$, we add the following children:

- for each child of $\tau_j$ which is a leaf witnessing some element of $K_{\vec{e}} \setminus \{\vec{e}\}$, we have a corresponding leaf of our larger process,
- for each child of $\tau_j$ which is an internal node of the process of type $K_{\vec{e}_j}$, we have a corresponding internal node,
- for each $i \leqslant k$, children with witnesses of type $\vec{d}^{+i}$.

To see that these are sufficient, consider our witnesses of type $\vec{d}$, and let $i$ be a coordinate so that there are infinitely many possible extensions of the witness at that coordinate. If $d_j = 1$ for all $j > i$, this means we can find a witness of type $d^{+i} \in K'$ and treat this as a leaf. Otherwise, consider the smallest $j > i$ with $d_j \neq 1$, and let $\vec{e} = \vec{d} \upharpoonright j$. The process of type $K_{\vec{e}}$ would normally have children giving witnesses of type $\vec{e}$, but we can instead ask for the $i$-th coordinate to extend the $i$-th coordinate of our existing witness, giving a witness of type $d^{+i}$.s

That is, we have a subprocess which either gives a leaf with witnesses of some type in $K'$, or we have witnesses to some longer node from one of the subprocess.

We now proceed inductively as follows. Take value $c'_j \leqslant c_j$, and we proceed by induction on $\sum_j c'_j$. Suppose that we are at a node where we have constructed a very large number of witnesses to nodes of height $c'_j$ to the process of type $K_{\vec{e}_j}$. (When $\sum_j c'_j = 0$, this is trivial, because we need nothing to witness a node of height 0.) We may apply the construction above; at each non-leaf, we have lost one witness to a node of height $c'_j$ for each $j$, but for a single $j$, have gotten a witness to a node of height $c'_j + 1$. By applying this a large number of times and using the pigeonhole principle, we can arrange for the same $j$ to be selected many times, so we obtain a large number of witnesses corresponding to a sequence $c'_1, \ldots, c'_j + 1, \ldots, c'_{m-1}$.

We repeat this until we reach nodes where we have a single witness to a node of height $c_j$ for each $j < m$. The construction above gives only leaves in that case, because the subprocesses have no non-leaf children. Therefore this gives us a construction of a process where each leaf gives witnesses of some type in $K'$. $\dashv$

Let $\vec{d}^{+}$ be the successor of $\vec{d}$ in the reverse lexicographic ordering. We show by strong induction that, having constructed a process of type $K_{\vec{e}}$ for each $\vec{e} \leqslant \vec{d}$, we also have a process of type $K_{\vec{d}^{+}}$. By the claim, we have a process of type $K' = K^+_{\vec{d}} \cup \bigcup_{\vec{e} \in Z(\vec{d})} (K_{\vec{e}} \setminus \{\vec{e}\})$, so it suffices to show that $K' \supseteq K_{\vec{d}^+}$.

Let $i_0$ be least so that $d_{i_0} < D_{i_0}$, so $\vec{d}^+ = \vec{d}^{+i_0}$. Note that $d^+_{i_0} > 1$. Suppose $\vec{f} \in K'$. If $\vec{f} \in K^+_{\vec{d}}$ then $\vec{f} = \vec{d}^{+i}$, and since $\vec{f} \in K$, $i \geqslant i_0$, so either $\vec{f} = \vec{d}^{+i_0}$, or $\vec{f} = (\vec{d}^+)^{+i}$ for some $i > i_0$, so $\vec{f} \in K_{\vec{d}^+}$.

Otherwise $\vec{f} \in K_{\vec{e}} \setminus \{\vec{e}\}$ for some $\vec{e} \in Z(\vec{d}) \setminus \{\vec{d}\}$. So $\vec{e} = \vec{d} \upharpoonright i$ for some $i$ with $d_i \neq 1$ and there is a $j > i$ with $\vec{f} = \vec{e}^{+j}$. Therefore $\vec{f} = \vec{d}^{+j}$. If $j < i_0$ then $\vec{f} \notin K$, so $j \geqslant i_0$, and therefore $\vec{f} \in K_{\vec{d}^+}$.

Therefore the claim gives us a process of type $K_{\vec{d}^+}$. Applying this repeatedly gives us a process of type $K_{(D_1,\dots,D_k)}$, which suffices to complete the proof. □

Combining these as before, we have:

**Theorem 3.7.** *There is a Turing ideal satisfying* $\mathbf{trRT^2_k}$ *for all* $k$ *and* **WKL** *but not* **SProdWQO**.

## 4. Separating SCAC

In this section we construct a computable instance $\preceq$ of **SCAC** and a Turing ideal $\mathcal{I}$ which has no solution to $\preceq$, but does satisfy both **ProdWQO** and **WKL**.

**Definition 4.1.** An **SCAC**-*requirement* is a requirement $R = (T, \{K_\alpha\}_{\sigma \in T}, \{d_\sigma\}_{\sigma \in T})$ with range $\{0,1\}$ and transitive in color 1.

Lemmata 2.8, 2.15 and 2.17 apply with $J = \{1\}$, $I = \{0,1\}$, so we have:

**Lemma 4.2.** *If $c$ satisfies all* **SCAC**-*requirements in $X$, taking $\prec$ to be the partial ordering so that $a \prec b$ iff $a < b$ and $c(a,b) = 1$, whenever $B$ is an $X$-computable infinite set, there exist $a,b,c,d \in B$, $a \prec b$, $c < d$ (so $d \nprec c$) and $c \nprec d$.*

**Lemma 4.3.** *If $c$ satisfies all* **SCAC**-*requirements in $X$ and $U$ is an infinite $X$-computable $\{0,1\}$-branching tree then there is an infinite branch $\Lambda$ so that $c$ satisfies all* **SCAC**-*requirements in $X \oplus \Lambda$.*

**Lemma 4.4.** *There is a computable stable $c : [\mathbb{N}]^2 \to \{0,1\}$ transitive in the color 1 satisfying every* **SCAC**-*requirement in $\emptyset$.*

In the lemma below, we associate a stable partial ordering $\preceq$ with a coloring with colors $0,1$ so that color 1 is transitive. In particular, we say that $\preceq$ satisfies an **SCAC**-requirement when the corresponding coloring does. So it remains to show:

**Lemma 4.5.** *Let $\mathcal{I}$ be a countable Turing ideal satisfying* **WKL**, *and suppose $\preceq$ satisfies every* **SCAC**-*requirement in any $X \in \mathcal{I}$ and $c : [\mathbb{N}]^2 \to \{0,1,2\}$ is a coloring in $\mathcal{I}$ with colors 1 and 2 transitive. Then there is an infinite set $S$ so that $c$ restricted to $S$ either omits the color 1 or omits the color 2 and $\preceq$ satisfies every* **SCAC**-*requirement in $X \oplus S$ for any $X \in \mathcal{I}$.*

Before we begin, we consider what makes solving **ProdWQO** more difficult than solving **ADS**; for simplicity, consider the stable version of both problems, where each point $a$ has a limit color $i$ so $c(a, x) = i$ for cofinitely many $x$. (Solving the unstable version is similar because we keep thinning to infinite subsets $X$ on which this holds for $x \in X$ and the finitely many values of $a$ which we have considered so far.) In an instance of **ProdWQO**, the only new phenomenon is that some points may have limit color 0; naïvely, this should only make the problem easier: such points can appear in either of the homogeneous sets we are constructing. The danger, of course, is that such points can also avoid making commitments (because the color is non-transitive), so if we rely on using them, we risk trying to use a point which appears to have limit color 0 only to have it later turn out to be homogeneously in the wrong color. As a result, we have to treat points which might have limit color 0, not as the best option—free points which we can choose to use anywhere—but as the worst option, points which might turn out to have either limit color.

To represent what limit colors might be, we introduce predictions.

**Definition 4.6.** If $S$ is a set, a *prediction* for $S$ is a function $r : S \to \{0, 1, 2\}$ such that if $a, b \in S$, $a < b$, and $c(a, b) = r(b) \neq 0$ then $r(a) = r(b)$. We say $r \leqslant r'$ if $\mathrm{dom}(r) \subseteq \mathrm{dom}(r')$ and whenever $r(a) \neq 0$, $r'(a) = r(a)$.

If $x \notin S$, $r^x_S$ is the prediction given by $r^x_S(a) = c(a, x)$.

A prediction represents a guess at the limit colors of the set $S$. We think of the comparison $r \leqslant r'$ as meaning that $r'$ is a later refinement of $r$. (More precisely, the fact we will use is that if we have $x, y, z$ with $r^x_S = r^y_S$, $c(x, z) = 1$, and $c(y, z) = 2$, then $r^x_S \leqslant r^z_S$.) Note that the comparison $\leqslant$ allows for points which appear to have limit color 0 in $r$ to turn out to have a non-zero limit color in a "later" prediction $r'$.

We begin from our usual situation: we have requirements $R^+$ and $R^-$, and we wish to build either a sequence $p$ omitting the color 1 and forcing $R^+$ or a sequence $q$ omitting the color 2 and forcing $R^-$.

The broad structure of our construction is familiar from Lemma 2.13. We will build a primary attempt at a solution, which we call the *backbone* (roughly analogous to the chains from that lemma, although, perhaps making the name inappropriate, the backbone will be branching, with each segment having two possible successor segments) and for each segment of the backbone we have an entire alternate solution which we call a *perpindicular* solution (roughly analogous to the partnered antichain from that lemma). One way in which this construction is more complicated is that the attempts are not single sequences. Each segment of the backbone consists of pairs of sequences (arranged so that we can obtain a sequence omitting the color 1 by taking the first element of a certain subset of pairs, and obtain a sequence omitting the color 2 by taking the second elements from a different subset). The perpindicular solutions are even more complicated: they are actually sequences of sets with the property that if we partition each set, we can find subsequences in these partitions which we can assemble to be the necessary witnessing sequences.

Most of the work is in the overall structure in which we play these two attempts against each other. Figure 13 illustrates this step. The segments in the perpindicular solutions are actually blocks of points which contain many possible sequences: the first segment is a set of points $S_1$ with the property that, for every possible prediction $r_{S_1}$ for $S_1$, there exist either:

- a sequence $p_{r_{S_1}} \subseteq S_1$ so that $c$ omits the color 1 on $p_{r_{S_1}}$, $p_{r_{S_1}}$ witnesses a node $\sigma_{r_{S_1}}$ of length 1 in $R^+$, and for $a \in p_{r_{S_1}}$, $r_{S_1}(a) \neq 1$, or
- a sequence $q_{r_{S_1}} \subseteq S_1$ so that $c$ omits the color 2 on $q_{r_{S_1}}$, $q_{r_{S_1}}$ witnesses a node $\tau_{r_{S_1}}$ of length 1 in $R^-$, and for $a \in p_{r_{S_1}}$, $r_{S_1}(a) \neq 2$.

Note that points with $r_{S_1}(a) = 0$ could be support either a $p_{r_{S_1}}$ or a $q_{r_{S_1}}$ (and must be allowed to since, for instance, the function $r_{S_1}$ which is constantly equal to 0 is a valid choice). We can begin our overall construction by simply searching for such an set; if we fail to find it, we use a technique from [5] to find a suitable infinite subset to continue our construction in.

We now look for a possible segment for the backbone. We restrain the first segment to 0 (this is the non-transitive color, so the first segment is inactive) while we look for a pair of sequences $(p^{\bullet\bullet}, q^{\bullet\bullet})$ where $p^{\bullet\bullet}$ witnesses a node of length 1 from $R^+$ and $q^{\bullet\bullet}$ witnesses a node of length 1 from $R^-$ (and $c$ omits the color 1 on $p^{\bullet\bullet}$ and the color 2 on $q^{\bullet\bullet}$; we will stop repeating this condition every time it applies). We use $p^{\bullet\bullet}$ and $q^{\bullet\bullet}$ to make a prediction on $S_1$: we also require that there is a single $r_{S_1}$ so that every $x \in p \cup q$ has $r^x_{S_1} = r_{S_1}$. We repeat this process many times, collecting a large number of such pairs, each with its own block of witnesses; we then pick a subset which all share the same nodes of length 1.

We would would like to choose a collection of pairs which all impose the same restraint $r_{S_1}$; this is too much to ask—the number of possible choices of $r_{S_1}$ is not determined by $R^+$ and $R^-$, so the number of pairs, and therefore the size of our tree, would depend on $|S_1|$, which is not permitted. Instead, we will choose a collection where $r_{S_1}$ may vary, but where the choices have certain commonalities; for the moment, however, we will pretend that we are actually able to choose a single value of $r_{S_1}$ which works for all our pairs $(p^{\bullet\bullet}_j, q^{\bullet\bullet}_j)$.

Suppose that $y > S_1 \cup p \cup q$ and $r_{S_1} \not\preccurlyeq r^y_{S_1}$; then there must either be some $a \in S_1$ with $r_{S_1}(a) = 1$ and $r^y_{S_1}(a) = 2$, or vice versa, so $r_{S_1}(a) = 2$ and $r^y_{S_1}(a) = 1$. Consider the former case (the latter is symmetric). Then when $z > y$, $c(y, z) = 2$ implies $c(a, z) = 2$, and therefore $c(x, z) \neq 1$ for all $x \in p$. Therefore, for every $z > y$, either $z$ can be used to build a sequence omitting 2 extending $y$, or a sequence omitting 1 extending $p$. A single such $y$ is not useful, but if we could find a new sequence $q'$ consisting of such $y$, we would then have a tool for extending our backbone.

If $r_{S_1} = r^y_{S_1}$ (the exact condition is more general), we can hope to use $y$ to extend the perpindicular solution. This leads to our next step: we make all these pairs $(p^{\bullet\bullet}, q^{\bullet\bullet})$ inactive. Our choice of $S_1$ guarantees us either a $p_{r_{S_1}}$ or a $q_{r_{S_1}}$ which is compatible with $r_{S_1}$ as described above. We cannot simply pick one, since it might be that we encounter many $y$ with $r_{S_1} < r^y_{S_1}$ and, say, $q_{r^y_{S_1}}$ is defined even though $p_{r_{S_1}}$ is defined. We must instead assemble the set of all possible choices. For technical reasons (because the construction of our tree can depend on the trees from $R^+$ and $R^-$, but cannot depend on the size of the witnessing set $S_1$ which gets found), we think of our choices in terms of $\sigma_{r_{S_1}}$ and $\tau_{r_{S_1}}$ rather than $p_{r_{S_1}}$ and $q_{r_{S_1}}$. Consider some $r'_{S_1} \geqslant r_{S_1}$ and say $\sigma_{r'_{S_1}}$ is defined above; for notational convenience, define $q_{r'_{S_1}} = \langle\rangle$ and $\tau_{r'_{S_1}} = \langle\rangle$. Dually, if $\tau_{r'_{S_1}}$ was defined in the conditions above, take $p_{r'_{S_1}} = \langle\rangle$ and $\sigma_{r'_{S_1}} = \langle\rangle$. Finally, we take $U_{r_{S_1}} = \{(\sigma_{r'_{S_1}}, \tau_{r'_{S_1}}) \mid r'_{S_1} \geqslant r_{S_1}\}$; this is the set of possible nodes we might have to deal with. (We can now point out that, instead of worrying about choosing $(p_j^{\bullet\bullet}, q_j^{\bullet\bullet})$ to share a common value of $r_{S_1}$, they can instead share a common value of $U_{r_{S_1}}$.)

We pick any pair $(\sigma, \tau) \in U_{r_{S_1}}$. This tells us how to restrain the first segment: if $\sigma \neq \langle\rangle$ then we restrain the first segment according to $\sigma$, and if $\tau \neq \langle\rangle$ then we restrain the first segment according to $\tau$. We now search for the next segment; we look for one of the following:

- a $p'$ such that:
  - $p'$ witnesses a node of length 1 from $R^+$,
  - there is an $r'_{S_1}$ so that, for every $y \in p'$, $r^y_{S_1} = r'_{S_1}$,
  - there is an $a \in S_1$ so that $r_{S_1}(a) = 2$ and $r^y_{S_1}(a) = 1$,
- a $q'$ such that:
  - $q'$ witnesses a node of length 1 from $R^-$,
  - there is an $r'_{S_1}$ so that, for every $y \in p'$, $r^y_{S_1} = r'_{S_1}$,
  - there is an $a \in S_1$ so that $r_{S_1}(a) = 1$ and $r^y_{S_1}(a) = 2$,
- a pair $(p', q')$ such that:
  - $p'$ witnesses a node of length 1 from $R^+$,
  - $q'$ witnesses a node of length 1 from $R^-$,
  - there is an $r'_{S_1}$ so that, for every $y \in p' \cup q'$, $r^y_{S_1} = r'_{S_1}$,
  - $r'_{S_1} \geqslant r_{S_1}$,
  - $(\sigma, \tau) \in U_{r'_{S_1}}$,
- a set $S_2$ so that
  - thre is an $r'_{S_1}$ so that, for every $y \in p'$, $r^y_{S_1} = r'_{S_1}$,
  - $r'_{S_1} \geqslant r_{S_1}$,
  - $(I, \sigma, \tau) \in U_{r'_{S_1}}$,
  - for every prediction $r_{S_{\leqslant 2}}$ for $S_{\leqslant 2} = S_1 \cup S_2$ such that $(\sigma, \tau) \in U_{r_{S_{\leqslant 2}} \upharpoonright S_1}$, there exists either
    * a sequence $p_{r_{S_{\leqslant 2}}}$ extending $p_{r_{S_{\leqslant 2}} \upharpoonright S_1}$, so that $c$ omits 1 on $p_{r_{S_{\leqslant 2}}}$, $p_{r_{S_{\leqslant 2}}}$ witnesses an immediate extension of $\sigma_{r_{S_{\leqslant 2}} \upharpoonright S_1}$ in $R^+$, and for every $a \in p_{r_{S_{\leqslant 2}}}$, $r_{S_{\leqslant 2}}(a) \neq 1$, or

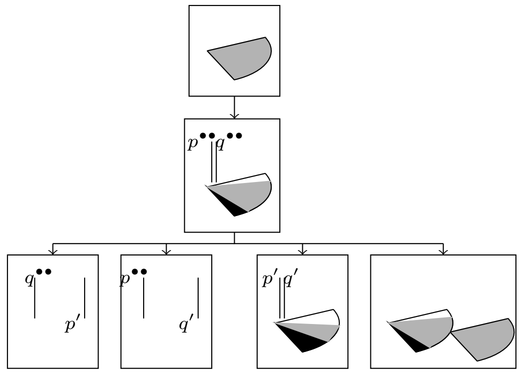


FIGURE 13.

* a sequence $q_{r_{S_{\leqslant 2}}}$ extending $q_{r_{S_{\leqslant 2}}\upharpoonright S_1}$, so that $c$ omits 2 on $q_{r_{S_{\leqslant 2}}}$, $q_{r_{S_{\leqslant 2}}}$ witnesses an immediate extension of $\tau_{r_{S_{\leqslant 2}}\upharpoonright S_1}$ in $R^-$, and for every $a \in q_{r_{S_{\leqslant 2}}}$, $r_{S_{\leqslant 2}}(a) \neq 2$.

In fact, the first two cases are combined in the proof below. If either of these cases happen, we have made progress towards building the backbone: we take either $(p_1, q_1) = (p', q^{\bullet\bullet})$ or $(p_1, q_1) = (p^{\bullet\bullet}, q')$ respectively, and have ensured that, for all sufficiently large $z$, either $c(y, z) \neq 1$ for $y \in p_1$, or $c(y, z) \neq 2$ for $z \in q_1$. (More precisely, we wait until we have many copies of $p'$ or many copies of $q'$, and then obtain many such pairs $(p_1, q_1)$, which we use as inputs to the inductive hypothesis which repeats the construction but where we have made progress towards constructing the backbone.) In the third case, we again wait until the third case has happened many times, and we then replace the pairs $p^{\bullet\bullet}, q^{\bullet\bullet}$ with the pair $p', q'$ and restart this process; in this case we have removed a triple $(\sigma, \tau)$ from $U_{r'_{S_1}}$, so the number of times we restart is bounded by the size of $R^+$ and $R^-$.

In the final case, we have made progress towards constructing the perpindicular solution. We now essentially restart the process we used after finding $S_1$: we discard the $(p^{\bullet\bullet}, q^{\bullet\bullet})$ and search for a new such pair, giving us a particular prediction $r_{S_{\leqslant 2}}$, which in turn tells us how to restrain the witnesses corresponding to $S_1$ and $S_2$, and allows us to continue this process.

This completes the description of how the two solutions interact. Let us now consider what happens if we simply keep extending the perpindicular solution. We get a sequence of sets, $S_1$, $S_2$, and so on. Taking $S_{\leqslant v} = \bigcup_{i \leqslant v} S_i$, the process above will give us a prediction $r_{S_{\leqslant v}}$. We will have a set $I \subseteq [1, v]$, which will be the stages at which the witness in $R^+$ was extended: if $I = \{n_1, \ldots, n_k\}$, we will have $p_1 \subseteq S_{n_1}$ witnessing a node $\sigma_1$ of length 1, $p_2 \subseteq S_{n_2}$ so that $p_1{}^\frown p_2$ witnesses an immediate extension $\sigma_2$ of $\sigma_1$, and so on. The stages after $n_k$ will witness extensions to $R^-$: we will have $q_1 \subseteq S_{n_k+1}$ witnessing a node $\tau_1$ of length 1, $q_2 \subseteq S_{n_k+2}$ so that $q_1{}^\frown q_2$ witnesses an immediate extension $\tau_2$ of $\tau_1$, and so on. (This corresponds to discarding our construction of an $R^-$ solution every time we extend the $R^+$ solution; we do this to prevent violations of the transitivity requirements.)

Finally, we consider how the backbone is constructed. Suppose we end up in one the first two cases of the four-way split above; for convenience, we assume the first case. Then we have many witnesses $q_j^{\bullet\bullet}$ and many witnesses $p'_j$ with the property that for every sufficiently large $y$, either:

- for every $j$ and every $x \in q_j^{\bullet}$, $c(y, x) \neq 2$, or
- for every $j$ and every $x \in p'_j$, $c(y, x) \neq 1$.

Then we can repeat this construction twice. The first, higher priority copy, will take place above the $q_j^{\bullet\bullet}$; for the sake of keeping names distinct, when we view these as inputs to the inductive hypothesis, we will call them $q_j^{\bullet}$. We modify this copy of the construction so that every time we look for a $q^{\bullet\bullet}$ or a $q'$, we choose some $q_j^{\bullet}$ and look for an extension of it (using a different $j$ each time we look for a new sequence). Simultaneously, we have a lower priority copy of our construction which takes place above the $p'_j$; again, we now call them $p_j^{\bullet}$. In this second copy of our construction, every time we look for a $p^{\bullet\bullet}$ or a $p'$, we choose some $p_j^{\bullet}$ to extend. In order to avoid transitivity conflicts between the two sides, every time the higher priority copy finds a witness, we discard the entire lower priority construction, starting it over with an entirely new batch of $p_j^{\bullet}$ (and therefore the number of witnesses $p'$ we must have found was quite large indeed).

*Proof.* We work with conditions $(p, q, r, X)$ such that:

- $c$ omits 1 on $p$,
- $c$ omits 2 on $q$,
- $r$ is a prediction for a finite set $S \supseteq p \cup q$,
- $a \in p$ implies $r(a) \neq 1$,
- $b \in q$ implies $r(b) \neq 2$,
- $X$ is an infinite set in $\mathcal{I}$,
- for every $x \in X$ we have $r^x_{\mathrm{dom}(r)} = r$.

A condition $(p', q', r', X')$ extends $(p, q, r, X)$ if $p \sqsubseteq p'$, $q \sqsubseteq q'$, $r \leqslant r'$, $X' \subseteq X$, and whenever $x \in (p' \backslash p) \cup (q' \backslash q)$, $x \in X$.

We say $(p, q, r, X)$ *forces* $R^+$ *on the omitting*-1-*side* if whenever $\Lambda$ is an infinite sequence with $p \sqsubset \Lambda$, $(\Lambda \backslash p) \subseteq X$, and $c(a, b) \neq 1$ for $a, b \in \Lambda$, $\preceq$ satisfies $R^+$ in $X \oplus \Lambda$. Similarly we say $(p, q, r, X)$ *forces* $R^-$ *on the omitting*-2-*side* if whenever $\Lambda$ is an infinite sequence with $q \sqsubset \Lambda$, $(\Lambda \backslash q) \subseteq X$, and $c(a, b) \neq 2$ for $a, b \in \Lambda$, $\preceq$ satisfies $R^-$ in $X \oplus \Lambda$.

It suffices to show

> $(*)$ Suppose $R^+$ are $R^-$ are requirements and $(p, q, r, X)$ is a condition. Then there is a condition $(p', q', r', X')$ extending $(p, q, r, X)$ which either forces $R^+$ on the omitting-1-side or forces $R^-$ on the omitting-2-side.

Let $(p, q, r, X)$, $R^+$, $R^-$ be given. We do not describe the tree explicitly, but again inductively describe processes that give rise to it. We will need to show the following generalization.

> Let $\sigma^\bullet \in T^+$ and $\tau^\bullet \in T^-$ be given. There is an $M$ so that whenever $p_1^\bullet, \ldots, p_M^\bullet, q_1^\bullet, \ldots, q_M^\bullet$ and $X^\bullet \subseteq X$ are given so that:
> - each $p_i^\bullet$ is an extension of $p$ omitting 1 which witnesses $\sigma^\bullet$,
> - each $q_i^\bullet$ is an extension of $q$ omitting 2 which witnesses $\tau^\bullet$,
> - for any $x \in \bigcup p_i^\bullet$ and $y \in X^\bullet$, $c(x, y) \neq 1$, and
> - for any $x \in \bigcup q_i^\bullet$ and $y \in X^\bullet$, $c(x, y) \neq 2$,
>
> there is a condition $(p', q', r', X')$ extending $(p, q, r, X)$ which either forces $R^+$ on the omitting-1-side or forces $R^-$ on the omitting-2-side.

We show this by reverse induction (the "main induction") on $(|\sigma^\bullet|, |\tau^\bullet|)$ in the lexicographic order. Here the $p_i^\bullet, q_i^\bullet$ represent prior progress towards the construction of the backbone; therefore showing the inductive step will involve showing that we either construct a perpindicular solution which gives the desired extension or extend the backbone, thereby reducing the problem to cases covered by the inductive hypothesis.

The base case, where either $\sigma^\bullet$ or $\tau^\bullet$ is a leaf is immediate with $M = 1$, since then either $(p_1^\bullet, q, r', X^\bullet)$ or $(p, q_1^\bullet, r', X^\bullet)$ is the desired extension (with $r' = r_S^y$ for $y \in X^\bullet$ and $S = \mathrm{dom}(r) \cup p_1^\bullet \cup q_1^\bullet$). Suppose the claim holds for all $\sigma', \tau'$ with either $\sigma' \supsetneq \sigma^\bullet$, or $\sigma' = \sigma^\bullet$ and $\tau' \supsetneq \tau^\bullet$.

Within this, we need another inductive construction (the "side induction"). We assume we have been given a collection of pairs $(p_i^\dagger, q_i^\dagger)$ for $i \leqslant K$ witnessing $\sigma^\dagger \in T^+$ and $\tau^\dagger \in T^-$ respectively; this will be arranged so they share blocks of witnesses in such a way that all pairs can be simultaneously active. Given $i \leqslant K$, we say $y$ is *$i$-extending* if for each $x \in p_i^\dagger$, $c(x, y) \neq 1$ and for each $x \in q_i^\dagger$, $c(x, y) \neq 2$. We assume we have also been given an infinite $X^\dagger \subseteq X^\bullet$ so that each $y \in X^\dagger$ is $i$-extending for at least one $i \leqslant K$. This represents previous stages of the perpindicular construction.

We can now begin the next stage of our construction. All $(p_i^\bullet, q_i^\bullet)$ are initially inactive, and the $(p_i^\dagger, q_i^\dagger)$ are all simultaneously active. We look for a finite set $S \subseteq X^\dagger$ so that there is an $r' \geqslant r$ with $\mathrm{dom}(r') \supseteq \bigcup_i p_i^\dagger \cup q_i^\dagger$ and so that, for each $y \in S$, $r^y_{\mathrm{dom}(r')} = r'$, and for every prediction $r_S$ on $S$ so that $r_S \cup r'$ is a prediction and every $i$ so that the $y$ in $S$ are $i$-extending (this depends on $r'$, so is independent of the choice of $y$), either:

- there is a $p_{i,r_S} \subseteq S$ and an immediate extension $\sigma_{i,r_S}$ of $\sigma^\dagger$ so that $r_S(a) \neq 1$ for $a \in p_{i,r_S}$ and $p_i^\dagger{}^\frown p_{i,r_S}$ omits 1 and witnesses $\sigma_{i,r_S}$ (in this case $q_{i,r_S} = q$ and $\tau_{i,r_S} = \langle\rangle$), or

- there is a $q_{i,r_S} \subseteq S$ and an immediate extension $\tau_{i,r_S}$ of $\tau^\dagger$ so that $r_S(a) \neq 2$ for $a \in q_{i,r_S}$ and $q_i^\dagger{}^\frown q_{i,r_S}$ omits 2 and witnesses $\tau_{i,r_S}$ (in this case $p_{i,r_S} = p_i^\dagger$ and $\sigma_{i,r_S} = \sigma$).

This is the general form of the search for the next segment of the perpindicular solution as described above. The key idea is that we extend a pair by extending one component or the other, and that for each pair $(p_i^\dagger, q_i^\dagger)$ we must extend it if we can—that is, if this pair is consistent with $r'$. Note that when we extend on the $q$ side we leave the $p$ side alone, while when we extend on the $p$ side, we discard the $q$ side; this is to ensure that we do not have overlapping transitivity requirements.

Suppose we never find such an $S$. We use the technique of Lemma 4.22 of [5]. Choose $r' \geqslant r$ with $\mathrm{dom}(r') \supseteq \bigcup_i p_i^\dagger \cup q_i^\dagger$ so that there are infinitely many $y \in X^\dagger$ with $r' = r^y_{\mathrm{dom}(r')}$; by passing to a subset, we may assume this holds for all $y \in X^\dagger$. Let $C$ be the non-empty set of $i$ so that all the $y \in X^\dagger$ are $i$-extending. On any initial segment $Y$ of $X^\dagger$, consider those partitions $Y = Y_1 \cup Y_2$ (equivalently, the prediction given by $r_Y(y) = b$ iff $y \in Y_b$) so that there is an $i \in C$ so that there do not exist either a $p_{i,r_Y} \subseteq Y_1$ or a $q_{i,r_Y} \subseteq Y_2$ as in the previous paragraph. Call such a partition a *bad partition* of $Y$. The bad partitions form a tree, since when $Y \subseteq Z = Z_1 \cup Z_2$, $Y \cap Z_1, Y \cap Z_2$ is a bad partition of $Y$. Suppose there is some $Y$ and some bad partition $Y = Y_1 \cup Y_2$ such that for every $Z \supseteq Y$, there is a bad partition $Z = Z_1 \cup Z_2$ with $Z_1 = Y_1$. Then, letting $X' = X^\dagger \backslash Y_1$, there must be some $i$ so that $(p, q_i^\dagger, r', X')$ forces $R^-$ on the omitting-2-side by forcing $\Theta^{X\oplus\Lambda}_{R^-;\tau^\dagger}$ to hold.

Suppose there is no such $Y = Y_1 \cup Y_2$; then whenever $Y = Y_1 \cup Y_2$ is a bad partition, there must be a $Z \supseteq Y$ so that any bad partition $Z = Z_1 \cup Z_2$ has either $Y_1 \nsubseteq Z_1$ or $Z_1 \supsetneq Y_1$. We describe a finitely branching tree of sets and, for each level of the tree, a number $n$, as follows. For each $n$, let $Z^n = X\dagger \cap [0, n)$. When $n$ corresponds to some level of the tree, the sets in that level will be exactly those $Z_1$ so that $Z^n = Z_1 \cup Z_2$ is a bad partition. 0 corresponds to the first level, so the only set in the first level is $\varnothing$. Suppose we have chosen $n$ to be some level of this tree. We wait for some $n'$ so that, for each $Z_1$ in the previous level, $Z^{n'} = Z_1 \cup (Z^{n'} \backslash Z_1)$ fails to be a bad partition. Then descendents of each $Z_1$ in the previous level are those $Z_1' \subsetneq Z_1$ so that $Z^{n'} = Z_1' \cup (Z^{n'} \backslash Z_1')$ is a bad partition. (Some nodes from the previous level may have no descendents.) Since $\mathcal{I}$ satisfies $\mathbf{WKL_0}$, we can choose an $X' \in \mathcal{I}$ which is the union of an infinite branch through this tree. There must be some $i$ so that $(p_i^\dagger, q, r', X')$ then forces $R^+$ on the not-1-side by forcing $\Theta^{X\oplus\Lambda}_{R^+;\sigma^\dagger}$ to hold.

So suppose we find the set $S$, and let $C$ be the non-empty set of $i$ so that all the $y \in S$ are $i$-extending. For each $r_S$, let $U_{r_S} = \{(\sigma_{i,r'_S}, \tau_{i,r'_S}) \mid r'_S \geqslant r_S, i \in C\}$; note that, although the number of possible values of $r_S$ depends on $|S|$, the number of possible values of $U_{r_S}$ is determined by the size of $T^+$

and $T^-$. We now repeatedly choose a single $(p_j^\bullet, q_j^\bullet)$, activate this pair, and look for a pair $(p_j^{\bullet\bullet}, q_j^{\bullet\bullet})$ so that:

- $p_j^\bullet \sqsubseteq p_j^{\bullet\bullet}$ and $q_j^\bullet \sqsubseteq q_j^{\bullet\bullet}$,
- $p_j^{\bullet\bullet}$ witnesses an immediate extension $\sigma_j^{\bullet\bullet}$ of $\sigma_j^\bullet$ and $q_j^{\bullet\bullet}$ witnesses an immediate extension $\tau_j^{\bullet\bullet}$ of $\tau_j^\bullet$,
- there is an $r_S^j$ so that, for all $x \in (p_j^{\bullet\bullet} \backslash p_j^\bullet) \cup (q_j^{\bullet\bullet} \backslash q_j^\bullet)$, $r_S^x = r_S^j$.

If, at any point, we fail to find such such a pair, we choose some $r_S$ so that there are infinitely many $x \in X^\dagger$ with $r_S^x = r_S$, let $X' \subseteq X^\dagger$ be the infinite subset of these $x$, and then either $(p, q_j^\bullet, r', X')$ or $(p_j^\bullet, q, r', X')$ is the desired extension (where $r' \geqslant r \cup r_S$).

So suppose we find these extensions for many values of $i$. We fix some number $M'$ (determined by the main inductive hypothesis) and wait until there is a set $L$ of indices, a $U$, and $\sigma^{\bullet\bullet}, \tau^{\bullet\bullet}$ so that:

- $|L| = M'$,
- for each $j \in L$, $U_{r_S^j} = U$, $\sigma_j^{\bullet\bullet} = \sigma^{\bullet\bullet}$, and $\tau_j^{\bullet\bullet} = \tau^{\bullet\bullet}$.

Since there are a fixed number of choices for $U$, $\sigma^{\bullet\bullet}$, and $\tau^{\bullet\bullet}$ (with the number determined by $R^+$ and $R^-$), if we are able to extend enough pairs then we find $L$.

We choose some $(\sigma', \tau') \in U$ and arrange the restraint according to this pair. We now look for three possible types of extensions simultaneously:

- we look for new pairs $(p_j^{\bullet\bullet}, q_j^{\bullet\bullet})$ as above where $(\sigma', \tau') \notin U_{r_S^j} \subsetneq U$,
- we look for pairs $(p'_j, q'_j)$ extending some $(p_j^\bullet, q_j^\bullet)$ as above so that, for every $j' \in L$, there is an $a \in S$ with $0 \neq r_S^j(a) \neq r_S^{j'}(a) \neq 0$,
- we follow the steps given by the side induction with the collection of $p_{i,r_S}, q_{i,r_S}$ such that $\sigma_{i,r_S} = \sigma'$ and $\tau_{i,r_S} = \tau'$.

Suppose we find none of these witnesses. Choose an $X' \subseteq X^\dagger$ and an $r_S$ so that, for each $y \in X'$, $r_S^y = r_S$. If $(\sigma', \tau') \in U_{r_S}$ then $X'$ suffices to carry out the side induction. If $U_{r_S} \subsetneq U$ then the failure to find pairs witnessing the first case implies that some $(p_j^\bullet, q, r', X')$ or $(p, q_j^\bullet, r', X')$ is the desired extension. If neither of these hold then for each $j \in L$ we cannot have $r_S \geqslant r_S^j$; therefore there is some $a$ so that $0 \neq r_S^j(a) \neq r_S(a) \neq 0$, and the failure to find pairs witnessing the second case again implies that some $(p_j^\bullet, q, r', X')$ or $(p, q_j^\bullet, r', X')$ is the desired extension.

If we find enough witnesses to the first case all sharing the same $U_{r_S^j} = U' \subsetneq U$ and the same $\sigma_j^{\bullet\bullet}$ and $\tau_j^{\bullet\bullet}$, we restart this portion with $U'$ replacing $U$. If we keep finding witnesses in the the third case, the side induction ensures that we finish.

Suppose we find a large number of witnesses to the second case (the number is quite large, but is still determined by the values from the main induction and $R^+$ and $R^-$); say these pairs are $(p'_j, q'_j)$ for $j \in R$. We can restrict to $L' \subseteq L$ and $R' \subseteq R$ so that either:

- for every $j \in L'$ and $j' \in R'$, there is an $a \in S$ with $r^j_S(a) = 1$ and $r^{j'}_S(a) = 2$, or
- for every $j \in L'$ and $j' \in R'$, there is an $a \in S$ with $r^j_S(a) = 2$ and $r^{j'}_S(a) = 1$.

Let us consider the first case. Then for any $y$, we either have $c(x, y) \neq 1$ for all $j \in L'$ and $x \in p^{\bullet\bullet}_j$, or $c(x, y) \neq 2$ for all $j \in R'$ and $x \in q'_j$. Partition $X^\dagger = X^\bullet_1 \cup X^\bullet_2$ where $X^\bullet_1$ is those $y$ so that, for every $j \in L'$ and $x \in p^{\bullet\bullet}_j$, $c(x, y) \neq 1$. We now take sufficiently many pairs $(p^{\bullet\bullet}_j, q)$ with $j \in L'$ and can apply the main induction as a subprocess with $X^\bullet_1$ (incrementing $u^+$ and letting $u^-$ reset to 0); we simultaneously take sufficiently many pairs $(p^\bullet_j, q'_j)$ with $j \in R'$ and apply the main induction again as a second subprocess with $X^\bullet_2$ (keeping $u^+$ the same and incrementing $u^-$). Every time the first subprocess finds witnesses to a new stage, we discard the second subprocess and start a new one with new witnesses from $R'$. Since one of $X^\bullet_1$ and $X^\bullet_2$ must be infinite, one of these subprocesses finds the necessary extension.

The second case is symmetric (using $(p^\bullet_j, q^{\bullet\bullet}_j)$ and $(p'_j, q)$ as witnesses to the inductive clauses).

This completes the inductive step for the side induction. In particular, we have shown that if we start with $K = 1$, $p^\dagger_1 = p$, $q^\dagger_1 = q$, $\sigma^\dagger = \tau^\dagger = \langle\rangle$, we can construct the desired extensions. This, in turn, completes the proof of the main induction. Applying the claim we have shown using the main induction to the pair $(0, 0)$ (with each $p^\bullet_i = p$, $q^\bullet_i = q$, $X^\bullet = X$), we obtain the desired extension. □

Combining these as before, we have:

**Theorem 4.7.** *There is a Turing ideal satisfying* **ProdWQO** *and* **WKL** *but not* **SCAC**.

## References


[1] P. A. Cholak, M. Giusto, J. L. Hirst, and C. G. Jockusch Jr. "Free sets and reverse mathematics". In: *Reverse mathematics 2001*. Vol. 21. Lect. Notes Log. Assoc. Symbol. Logic, La Jolla, CA, 2005, pp. 104–119.

[2] P. Cholak, A. Marcone, and R. Solomon. "Reverse mathematics and the equivalence of definitions for well and better quasi-orders". In: *J. Symbolic Logic* 69.3 (2004), pp. 683–712. ISSN: 0022-4812.

[3] H. M. Friedman. "FOM:53:free sets and reverse math and FOM:54:recursion theory and dynamics." Available at `https://www.cs.nyu.edu/pipermail/fom/`..

[4] D. R. Hirschfeldt and R. A. Shore. "Combinatorial principles weaker than Ramsey's theorem for pairs". In: *J. Symbolic Logic* 72.1 (2007), pp. 171–206. ISSN: 0022-4812.

[5] M. Lerman, R. Solomon, and H. Towsner. “Separating principles below Ramsey’s Theorem for pairs”. In: *Journal of Mathematical Logic* (), p. 1350007. eprint: `http://www.worldscientific.com/doi/pdf/10.1142/S0219061313500074`.

[6] S. Murakami, T. Yamazaki, and K. Yokoyama. “On the Ramseyan factorization theorem”. In: *Language, life, limits*. Vol. 8493. Lecture Notes in Comput. Sci. Springer, Cham, 2014, pp. 324–332.

[7] L. Patey. “The weakness of being cohesive, thin, or free in reverse mathematics”. to appear, Israel J. Math.

[8] L. Patey. “Partial Orders and Immunity in Reverse Mathematics”. In: *Pursuit of the Universal: 12th Conference on Computability in Europe, CiE 2016, Paris, France, June 27 - July 1, 2016, Proceedings*. Ed. by A. Beckmann, L. Bienvenu, and N. Jonoska. Cham: Springer International Publishing, 2016, pp. 353–363. ISBN: 978-3-319-40189-8.

[9] S. G. Simpson. *Subsystems of second order arithmetic*. Second. Perspectives in Logic. Cambridge: Cambridge University Press, 2009, pp. xvi+444. ISBN: 978-0-521-88439-6.

[10] W. Wang. “Some logically weak Ramseyan theorems”. In: *Adv. Math.* 261 (2014), pp. 1–25. ISSN: 0001-8708.



Department of Mathematics, University of Pennsylvania, 209 South 33rd Street, Philadelphia, PA 19104-6395, USA

*E-mail address*: `htowsner@math.upenn.edu`

*URL*: `http://www.math.upenn.edu/~htowsner`