Computable Categoricity, and Topology in Reverse Mathematics

Java Darleen Villano, PhD

University of Connecticut, 2025

Abstract

We say that a computable structure $\mathcal{A}$ is computably categorical if for every computable copy $\mathcal{B}$, there exists a computable isomorphism $f : \mathcal{A} \to \mathcal{B}$. This notion can be relativized to a degree $\mathbf{d}$ by saying that a computable structure $\mathcal{A}$ is computably categorical relative to $\mathbf{d}$ if for every $\mathbf{d}$-computable copy $\mathcal{B}$ of $\mathcal{A}$, there exists a $\mathbf{d}$-computable isomorphism $f : \mathcal{A} \to \mathcal{B}$. A key part of this thesis is to study the behavior of this notion of categoricity in the computably enumerable (c.e.) degrees. We show that it is badly behaved in the c.e. degrees by extending a previously known result by Downey, Harrison-Trainor, and Melnikov in [9] to partial orders of c.e. degrees (Theorem 1.1.10). We also show that using largely the same techniques alongside a standard construction of minimal pairs, we can embed a four-element diamond lattice into the c.e. degrees in the style of Theorem 1.1.10.

We then apply some of the techniques used in this thesis to study the behavior of this notion in the context of generic degrees. Additionally, we show that several classes of structures admit a computable example that witnesses the pathological behavior of categoricity relative to a degree as seen in Theorem 1.1.10.

Lastly, in the context of reverse mathematics, we investigate the reverse mathematical strength of a topological principle named $\mathsf{wGS}^{\mathrm{cl}}$, a weakened version of the Ginsburg-Sands theorem which states that every infinite topological space contains one of the following five topologies as a subspace, with $\mathbb{N}$ as the underlying set: discrete, indiscrete, cofinite, initial segment, or final segment.

Computable Categoricity, and Topology in Reverse Mathematics

Java Darleen Villano

B.A. Mathematics, University of California, Berkeley, 2019

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2025

Doctor of Philosophy Dissertation

Computable Categoricity, and Topology in Reverse Mathematics

Presented by

Java Darleen Villano, B.A.

Approved by

Major Advisor: David Reed Solomon

Major Advisor: Damir D. Dzhafarov

Associate Advisor: Karen Lange

University of Connecticut

2025

Acknowledgements

I would like to begin my acknowledgments with a very heartfelt thank you to my advisors, Reed Solomon and Damir Dzhafarov. To do and write mathematics is extremely challenging, but I have been able to overcome all challenges so far thanks to the patience, guidance, and kindess from Reed and Damir. They have given me bountiful support for whatever academic interests I picked up, as well as many resources to become successful as a PhD student. They have also given me countless interesting mathematical questions and pursuits, most which ended up in this dissertation. In particular, I would also like to express further gratitude to Reed for carefully checking my drafts and helping me develop my mathematical writing. Lastly, I'd like to again thank both Reed and Damir for all the advice, both mathematical and not, they've shared with me over the years.

Besides Reed and Damir, I'd also like to thank my fellow colleagues for their helpful conversations or friendly chats whenever I run into them at conferences or meetings: Nikolay Bazhenov, Wesley Calvert, Heidi Benham, Andrew DeLapo, Johanna Franklin, Jun Le Goh, David Gonzalez, Emma Gruner, Liling Ko, Alex Kruckman, Karen Lange, Elvira Mayordomo, Russell Miller, Christopher Porter, Diego Rojas, Dino Rossegger, Dan Turetsky, Ted Slaman, and Andrea Volpi. I know that I can be very shy and not too talkative, so I appreciate all the efforts over the years to make sure that I am included at any social function at a conference or the mathematical conversations happening during the breaks between talks. Moreover, coming up with ideas for the work I've done the past few years probably would have been impossible without some of the conversations I've had with the people above.

Beyond the mathematics, I'd also like to thank the following people at UConn (formerly or currently) for all the help and mentorship they've given me: Rachel D'Antonio, Sougata Dhar, Masha Gordina, Katie Hall, Dave McArdle, and Monique Roy. Without their help, I would have not been able to fulfill my teaching duties to the many students I've had over the past few years. I'd also like to thank them for all the administrative help and advice they have given me.

Next, I'd like to also thank my cohort: Rachel Bailey, Ben Oltsik, Kim Savinon, Ryan Schroeder, Erik Wendt, and Ben York. It was very fun and memorable to have shared Monteith 422 with you all for the first five years together. Additionally, I felt that with you all, I had a space where I belonged in the UConn math department, which helped me stay productive and focused on research projects.

Lastly, I'd like to thank my family and friends for their continued support and encouragement. Although several friends have told me that they do not quite understand what I do, the fact that they continue to support me anyways means a lot to me. Finally, I'd like to thank Aidan Backus, my husband, who also happens to be a mathematician. Thank you so much for always supporting me and for understanding the challenges in doing mathematics.

Table of Contents

Chapter 1

Computable categoricity relative to a c.e. degree

1.1 Introduction

1.1.1 Preliminaries

In computable structure theory, we are interested in effectivizing model theoretic notions and constructions. For a general background on computable structure theory, see Downey [8] and Ash and Knight [2]. In particular, many people have examined the complexity of isomorphisms between structures within the same isomorphism type. We restrict ourselves to countable structures in a computable language and assume their domain is ω.

Definition 1.1.1. A computable structure $\mathcal{A}$ is **computably categorical** if for any computable copy $\mathcal{B}$ of $\mathcal{A}$, there exists a computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

There are many known examples of computably categorical structures including computable linear orderings with only finitely many adjacent pairs [26], computable fields of finite transcendence degree [11], and computable ordered graphs of finite rank [18]. In each of these examples, the condition given turns out to be both necessary and sufficient for computable categoricity.

We are also interested in studying relativizations of computable categoricity. The most studied relativization of this notion is relative computable categoricity.

Definition 1.1.2. A computable structure $\mathcal{A}$ is **relatively computably categorical** if for any copy $\mathcal{B}$ of $\mathcal{A}$, there exists a $\mathcal{B}$-computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

We can think of this relativization as relativizing categoricity to *all* degrees at once since we do not fix the complexity of our copies of $\mathcal{A}$. If a structure $\mathcal{A}$ is relatively computably categorical, then it is computably categorical. The converse does not hold in general, but often holds for structures where there is a purely algebraic characterization of computable categoricity. In particular, the examples of computable categorical structures listed above are also relatively computably categorical.

The connection between a purely algebraic characterization of computable categoricity and the equivalence of computable categoricity and relative computable categoricity was clarified by the following result which was independently discovered by Ash, Knight, Manasse, and Slaman [3], and Chisholm [6].

Theorem 1.1.3 (Ash et al., Chisholm)**.** A structure $\mathcal{A}$ is relatively computably categorical if and only if it has a formally Σ^0_1 Scott family.

In [17], Gončarov used an enumeration result of Selivanov [28] to give the first example of a computably categorical structure that is not relatively computably categorical. Later, in [16], Gončarov proved that if a computable structure has a single 2-decidable copy, then it is relatively computably categorical. Kudinov [21] constructed an example showing that the assumption of 2-decidability could not be lowered to 1-decidability.

Theorem 1.1.4 (Gončarov)**.** If a structure is computably categorical and its $\forall\exists$ theory is decidable, then it is relatively computably categorical.

We expand from computable isomorphisms by allowing a fixed number of jumps.

Definition 1.1.5. For a computable ordinal α, we say that a computable structure $\mathcal{A}$ is **Δ^0_α-categorical** if for any computable copy $\mathcal{B}$ of $\mathcal{A}$, there is a Δ^0_α-computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

There are relatively few known characterizations of Δ^0_α-categoricity within particular classes of structures, and what is known often requires extra computability theoretic assumptions. For example, McCoy [24] considered computable Boolean algebras for which the set of atoms and the set of atomless elements were computable in at least one computable copy. He proved that such a Boolean algebra is Δ^0_2-categorical if and only if it is a finite direct sum of atoms, 1-atoms, and atomless elements.

We can relativize Δ^0_α-categoricity in a similar way as with computable categoricity.

Definition 1.1.6. A computable structure $\mathcal{A}$ is **relatively Δ^0_α-categorical** if for any copy $\mathcal{B}$ of $\mathcal{A}$, there is a $\Delta^0_\alpha(\mathcal{B})$-computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

This notion also has a nice syntactic characterization, obtained by relativizing Theorem 1.1.3.

Theorem 1.1.7 (Ash et al., Chisholm)**.** A computable structure $\mathcal{A}$ is relatively Δ^0_α-categorical if and only if it has a formally Σ^0_α Scott family.

The extra computability assumptions needed to characterize Δ^0_α-categorical structures often disappear in the relativized version. McCoy [24] showed that a computable Boolean algebra is relatively Δ^0_2-categorical if and only if it is a finite direct sum of atoms, 1-atoms, and atomless elements, with no additional assumptions on the computable copy. He obtained similar results for linear orders in [24] and for Δ^0_3-categoricity in [25]. There are also examples of structures where plain and relativized Δ^0_α-categoricity coincide. In [4], Bazhenov proved that a computable Boolean algebra $\mathcal{B}$ is Δ^0_2-categorical if and only if it is also relatively Δ^0_2-categorical.

1.1.2 Categoricity relative to a degree

In this chapter, we focus on a different relativization of computable categoricity.

Definition 1.1.8. Let $\mathbf{d}$ be a Turing degree. A structure $\mathcal{A}$ is **computably categorical relative to d** if and only if for all $\mathbf{d}$-computable copies $\mathcal{B}$ of $\mathcal{A}$, there is a $\mathbf{d}$-computable isomorphism $f : \mathcal{A} \to \mathcal{B}$.

Using a relativized version of Gončarov's [16] result and Theorem 1.1.3, Downey, Harrison-Trainor, and Melnikov showed that if a computable structure $\mathcal{A}$ is computably categorical relative to a degree $\mathbf{d} \geq \mathbf{0}''$, then $\mathcal{A}$ is computably categorical relative to *all* degrees above $\mathbf{0}''$. In contrast, they also showed that being computably categorical relative to a degree is not a monotonic property below $\mathbf{0}'$.

Theorem 1.1.9 (Downey, Harrison-Trainor, Melnikov)**.** There is a computable structure $\mathcal{A}$ and c.e. degrees $0 = \mathbf{d}_0 < \mathbf{e}_0 < \mathbf{d}_1 < \mathbf{e}_1 < \ldots$ such that

(1) $\mathcal{A}$ is computably categorical relative to $\mathbf{d}_i$ for each i,

(2) $\mathcal{A}$ is not computably categorical relative to $\mathbf{e}_i$ for each i,

(3) $\mathcal{A}$ is relatively computably categorical to $\mathbf{0}'$.

We extend this result to partial orders of c.e. degrees, showing the full extent of pathological behavior for this relativization of categoricity underneath $\mathbf{0}'$.

Theorem 1.1.10. Let $P = (P, \leq)$ be a computable partially ordered set and let $P = P_0 \sqcup P_1$ be a computable partition. Then, there exists a computable computably categorical directed graph $\mathcal{G}$ and an embedding h of P into the c.e. degrees where $\mathcal{G}$ is computably categorical relative to each degree in $h(P_0)$ and is not computably categorical relative to each degree in $h(P_1)$.

The proof is a priority construction on a tree of strategies, using several key ideas from the proof of Theorem 1.1.9 in [9] along with some new techniques. In section 1.2, we introduce informal descriptions for the strategies we need to satisfy our requirements for the construction and discuss important interactions between certain strategies. In section 1.3, we detail the formal strategies, state and prove auxiliary lemmas about our construction, and state and prove the main verification lemma.

1.2 Informal strategies for Theorem 1.1.10

To prove Theorem 1.1.10, we have four goals to achieve within our construction, giving us four types of requirements to satisfy. In this section, we give informal descriptions of the strategies needed to satisfy each requirement in isolation, and then describe the interactions which arise when we employ these strategies together.

1.2.1 Embedding P into the c.e. degrees

We embed the poset P into the c.e. degrees in a standard way by constructing an independent family of uniformly c.e. sets A_p for $p \in P$. We fix the following notation:

$$\overline{D_p} := \bigoplus_{q \neq p} A_q.$$

For each $p \in P$, we ensure that $A_p \not\leq_T \overline{D_p}$. The image of p will be the c.e. set $D_p = \bigoplus_{q \leq p} A_q$. Because the A_p are independent, our embedding is order-preserving, i.e., $p \leq q$ in P if and only if $D_p \leq_T D_q$.

For each $p \in P$ and $e \in \omega$, we define the **independence requirement**:

$$N_e^p : \Phi_e^{\overline{D_p}} \neq A_p.$$

In order to satisfy an N_e^p requirement in isolation, we use the following N_e^p-strategy. Let α be an N_e^p-strategy. When α is first eligible to act, it picks a large number x_α. Once x_α is defined, α checks if $\Phi_e^{\overline{D_p}}(x_\alpha)[s] \downarrow= 0$. If not, α takes no action at stage s. If $\Phi_e^{\overline{D_p}}(x_\alpha)[s] \downarrow= 0$, then α enumerates x_α into A_p and preserves this computation by restraining

$\overline{D_p} \upharpoonright (\text{use}(\Phi_e^{\overline{D_p}}(x_\alpha)) + 1)$.

Notice that if we never see that $\Phi_e^{\overline{D_p}}(x_\alpha) \downarrow= 0$, then either $\Phi_e^{\overline{D_p}}(x_\alpha) \uparrow$ or $\Phi_e^{\overline{D_p}}(x_\alpha) \downarrow\neq 0$, and in either case, the value of $\Phi_e^{\overline{D_p}}(x_\alpha)$ will not be equal to $A_p(x_\alpha) = 0$ and so we meet the N_e^p requirement. Otherwise, at the first stage s for which $\Phi_e^{\overline{D_p}}(x_\alpha)[s] \downarrow= 0$, we enumerate x_α into A_p and restrain $\overline{D_p}$ below $\text{use}(\Phi_e^{\overline{D_p}}(x_\alpha)) + 1$. In this case we have that $\Phi_e^{\overline{D_p}}(x_\alpha) \downarrow= 0 \neq 1 = A_p(x_\alpha)$, and so we satisfy N_e^p.

1.2.2 Making $\mathcal{G}$ computably categorical

We will build $\mathcal{G}$ in stages. At stage $s = 0$, we set $\mathcal{G} = \emptyset$. Then, at stage $s > 0$, we add two new connected components to $\mathcal{G}[s]$ by adding the root nodes a_{2s} and a_{2s+1} for those components, and attaching to each node a 2-loop (a cycle of length 2). We then attach a $(5s+1)$-loop to a_{2s} and a $(5s+2)$-loop to a_{2s+1}. This gives us the configuration of loops:

$$a_{2s} : 2, 5s+1$$

$$a_{2s+1} : 2, 5s+2.$$

The connected component consisting of the root node a_{2s} with its attached loops will be referred to as the **2sth connected component** of $\mathcal{G}$. During the construction, we might add more loops to connected components of $\mathcal{G}$, which causes them to have one of the two following configurations:

$$a_{2s} : 2, 5s+1, 5s+3$$

$$a_{2s+1} : 2, 5s+1, 5s+4$$

or

$$a_{2s} : 2, 5s+1, 5s+2, 5s+3$$

$$a_{2s+1} : 2, 5s+1, 5s+2, 5s+4.$$

The idea behind adding these loops is to uniquely identify each connected component of $\mathcal{G}$. In all configurations above, there is only one way to match the components in $\mathcal{G}$ with components in a computable graph in order to define an embedding.

To make $\mathcal{G}$ computably categorical, we attempt to build an embedding of $\mathcal{G}$ into each computable directed graph. For each index e, let $\mathcal{M}_e$ be the (partial) computable graph with domain ω such that $E(x,y) \iff \Phi_e(x,y) = 1$ and $\neg E(x,y) \iff \Phi_e(x,y) = 0$. If Φ_e is not total, then $\mathcal{M}_e$ will not be a computable graph, but we will attempt to embed $\mathcal{G}$ into $\mathcal{M}_e$ anyway since we cannot know whether Φ_e is total or not. So we have the following requirement for each $e \in \omega$.

$$S_e : \text{if } \mathcal{G} \cong \mathcal{M}_e, \text{ then there exists a computable isomorphism } f_e : \mathcal{G} \to \mathcal{M}_e$$

To satisfy each S_e requirement in isolation, we have the following strategy. Let α be an S_e-strategy. When α is first eligible to act, it sets its parameter $n_\alpha = 0$ and defines its current map f_α from $\mathcal{G}$ into $\mathcal{M}_e$ to be empty. For the rest of this description, let $n = n_\alpha$. This parameter will keep track of the connected components that α is attempting to match between $\mathcal{G}$ and $\mathcal{M}_e$, and will be incremented by 1 only when we find copies of the $2n$th and $(2n+1)$st connected components of $\mathcal{G}$. Suppose the map $f_\alpha[s-1]$ matches up the $2m$th and $(2m+1)$st components of $\mathcal{G}[s-1]$ and $\mathcal{M}_e[s-1]$ for all $m < n$. At future stages, α checks whether $\mathcal{M}_e[s]$ contains isomorphic copies of the $2n$th and $(2n+1)$st components in $\mathcal{G}[s]$. If not, α takes no additional action at stage s and retains the parameter n and the map f_α. If $\mathcal{M}_e[s]$ contains copies of these components, then α extends the map by matching those components in $\mathcal{G}[s]$ and $\mathcal{M}_e[s]$, and it increments the value of n by 1.

If α finds copies of the $2n$th and $(2n+1)$st components of $\mathcal{G}$ for every n, then f_α will be a computable embedding of $\mathcal{G}$ into $\mathcal{M}_e$. Because of the form of $\mathcal{G}$, if $\mathcal{M}_e \cong \mathcal{G}$, then f_α will be a partial embedding which can be extended computably to a computable embedding on all of $\mathcal{G}$, satisfying the S_e requirement. Otherwise, there exists some n such that the $2n$th and $(2n+1)$st components of $\mathcal{G}$ were never matched, and so $\mathcal{G}$ and $\mathcal{M}_e$ cannot be isomorphic and so we trivially satisfy S_e.

1.2.3 Being computably categorical relative to a degree

In this construction, we want to define computations using a D_p-oracle that can be destroyed later by enumerating numbers into A_p. We achieve this by setting the use of the D_p-computation to be $\langle u, p\rangle$. Enumerating u into A_p causes $\langle u, p\rangle$ to enter D_p, destroying the associated computation.

For each $p \in P_0$, we ensure $\mathcal{G}$ is computably categorical relative to D_p. Let $\mathcal{M}_i^{D_p}$ be the (partial) D_p-computable directed graph with domain ω and edge relation given by $\Phi_i^{D_p}$. Since each D_p is c.e., we define the following terms to keep track of certain finite subgraphs which appear and remain throughout our construction.

Definition 1.2.1. Let C_0 and C_1 be isomorphic finite distinct subgraphs of $\mathcal{M}_i^{D_p}[s]$. The **age of $\mathbf{C_0}$** is the least stage $t \leq s$ such that all edges in C_0 appear in $\mathcal{M}_i^{D_p}[t]$, denoted by $\text{age}(C_0)$. We say that $\mathbf{C_0}$ is **older than $\mathbf{C_1}$** when $\text{age}(C_0) \leq \text{age}(C_1)$.

We say that C_0 is the **oldest** if for all finite distinct subgraphs $C \cong C_0$ of $\mathcal{M}_i^{D_p}[s]$, $\text{age}(C_0) \leq \text{age}(C)$.

Definition 1.2.2. Let $C_0 = \langle a_0, a_1, \ldots, a_k\rangle$ and $C_1 = \langle b_0, b_1, \ldots, b_k\rangle$ be isomorphic finite distinct subgraphs of $\mathcal{M}_i^{D_p}[s]$ with $a_0 < a_1 < \cdots < a_k$ and $b_0 < b_1 < \cdots < b_k$. We say that $C_0 <_{\text{lex}} C_1$ if for the least j such that $a_j \neq b_j$, $a_j < b_j$.

We say that C_0 is the **lexicographically least** if for all finite distinct subgraphs $C \cong C_0$ of $\mathcal{M}_i^{D_p}[s]$, $C_0 <_{\text{lex}} C$.

If $\mathcal{G} \cong \mathcal{M}_i^{D_p}$, then we need to build a D_p-computable isomorphism between these graphs. To achieve this, we meet the following requirement for each $i \in \omega$.

$$T_i^p : \text{if } \mathcal{G} \cong \mathcal{M}_i^{D_p}, \text{ then there exists a } D_p\text{-computable isomorphism } g_i^{D_p} : \mathcal{G} \to \mathcal{M}_i^{D_p}$$

The strategy to satisfy each T_i^p requirement in isolation is similar to the S_e-strategy, with some additional changes. Since the graphs are D_p-computable, embeddings defined by a T_i^p-strategy may become undefined later when small numbers enter D_p. Enumerations into

D_p also cause changes in $\mathcal{M}_i^{D_p}$ such as disappearing edges. We will show in the verification that if $\mathcal{G} \cong \mathcal{M}_i^{D_p}$, "true" copies of components from $\mathcal{G}$ will eventually appear and remain in $\mathcal{M}_i^{D_p}$ (and thus become the oldest finite subgraph which is isomorphic to a component in $\mathcal{G}$), and so our T_i^p-strategy below will be able to define the correct D_p-computable isomorphism between the two graphs.

Let α be a T_i^p-strategy. When α is first eligible to act, it sets its parameter $n_\alpha = 0$ and defines $g_\alpha^{D_p}$ to be the empty map. Once α has defined n_α, then at the previous stage $s-1$ (or the last α-stage in the full construction), we have the following situation:

- For each $m < n_\alpha$, $g_\alpha^{D_p}[s-1]$ maps the $2m$th and $(2m+1)$st components of $\mathcal{G}[s-1]$ to isomorphic copies in $\mathcal{M}_i^{D_p}[s-1]$.

- For $m < n_\alpha$, let l_m be the maximum $\Phi_i^{D_p}[s-1]$-use for the loops in the copies in $\mathcal{M}_i^{D_p}[s-1]$ for the $2m$th and $(2m+1)$st components in $\mathcal{G}$. We can assume that if $m_0 < m_1 < n_\alpha$, then $l_{m_0} < l_{m_1}$.

- For $m < n_\alpha$, let $\langle u_m, p\rangle$ be the $g_\alpha^{D_p}[s-1]$-use for the mapping of the $2m$th and $(2m+1)$st components of $\mathcal{G}$. This use will be constant for all elements in these components.

- By construction, we will have that $l_m < u_k \leq \langle u_k, p\rangle$ for all $m \leq k < n_\alpha$.

Suppose α is acting at stage s and has already defined n_α. We first check whether numbers have been enumerated into D_p that injure the loops in $\mathcal{M}_i^{D_p}[s-1]$ given by $\Phi_i^{D_p}[s-1]$. If not, then we keep the current value of n_α and skip ahead to the next step. If so, then let k be the least such that some number $x \leq l_k$ was enumerated into D_p. The loops in $\mathcal{M}_i^{D_p}[s]$ in the copies of the $2k$th and $(2k+1)$st components of $\mathcal{G}$ have been injured, and so may have disappeared. We have that $x \leq \langle u_m, p\rangle$ for all $k \leq m < n_\alpha$, and so our map $g_\alpha^{D_p}[s]$ is now undefined on all the $2m$th and $(2m+1)$st components for $k \leq m < n_\alpha$. So, α redefines $n_\alpha = k$ to find new images for the $2k$th and $(2k+1)$st components in $\mathcal{G}[s]$.

Second, we check to see if there is a $j < k$ and a new $x \in D_p$ such that $l_j < x < \langle u_j, p\rangle$. By minimality of the value k above, we know that $l_j < x$ for all $j < k$ and $x \in D_p[s] \setminus D_p[s-1]$.

For each such j, our map $g_\alpha^{D_p}[s-1]$ has been injured on the $2j$th and $(2j+1)$st components of $\mathcal{G}$, but the loops in the copies of those components in $\mathcal{M}_i^{D_p}[s]$ remain intact. Therefore, we define $g_\alpha^{D_p}[s]$ on these components with oracle $D_p[s]$ to be equal to $g_\alpha^{D_p}[s-1]$. Furthermore, we keep the same use for $g_\alpha^{D_p}[s]$ on these components. This will ensure that injury of this type happens only finitely often.

Third, we check whether we can extend $g_\alpha^{D_p}[s-1]$ to the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}[s]$. Search for isomorphic copies in $\mathcal{M}_i^{D_p}[s]$ of these components. If there are multiple copies in $\mathcal{M}_i^{D_p}[s]$, choose the oldest such copy to map to, and if there are multiple equally old copies, choose the lexicographically least oldest copy. If there are no copies in $\mathcal{M}_i^{D_p}[s]$, then keep the value of n_α the same and $g_\alpha^{D_p}$ unchanged and let the next requirement act. Otherwise, extend $g_\alpha^{D_p}[s-1]$ to $g_\alpha^{D_p}[s]$ to include the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$ and set the use to be $\langle u_{n_\alpha}, p\rangle$ where u_{n_α} is large (and so $u_{n_\alpha} > l_k$ for all $k \leq n_\alpha$). Increment n_α by 1 and go to the next requirement.

If $\mathcal{G} \cong \mathcal{M}_i^{D_p}$, then for each n, eventually the real copies of the $2n$th and $(2n+1)$st components of $\mathcal{G}$ will appear and remain forever in $\mathcal{M}_i^{D_p}$. Moreover, they will eventually be the oldest and lexicographically least copies in $\mathcal{M}_i^{D_p}$. Let l_n be the maximum true D_p-use on the edges in the loops in these $\mathcal{M}_i^{D_p}$ components. At this point, we will define $g_\alpha^{D_p}$ correctly on these components with a large use $\langle u_n, p\rangle$. Since $l_n < \langle u_n, p\rangle$, at most finitely many numbers enter D_p from the interval $(l_n, \langle u_n, p\rangle]$, but each time this happens, we define our map $g_\alpha^{D_p}$ to remain the same with the same use on the new oracle. Therefore, eventually our map $g_\alpha^{D_p}$ is never injured again on the $2n$th and $(2n+1)$st components. It follows that if $G \cong \mathcal{M}_i^{D_p}$, then $g_\alpha^{D_p}$ will be an embedding of $\mathcal{G}$ into $\mathcal{M}_i^{D_p}$ which will be an isomorphism by the structure of $\mathcal{G}$.

1.2.4 Being not computably categorical relative to a degree

Finally, for each $q \in P_1$ we want to make $\mathcal{G}$ not computably categorical relative to the c.e. set D_q. To achieve this, we build a D_q-computable copy $\mathcal{B}_q$ of $\mathcal{G}$ such that for all $e \in \omega$, the D_q-computable map $\Phi_e^{D_q} : \mathcal{G} \to \mathcal{B}_q$ is not an isomorphism. The graph $\mathcal{B}_q$ will be built globally.

Similarly to $\mathcal{G}$, we build the directed graph $\mathcal{B}_q$ in stages. At stage $s = 0$, we set $\mathcal{B}_q = \emptyset$. At stage $s > 0$, we add root nodes b^q_{2s} and b^q_{2s+1} to $\mathcal{B}_q$ and attach to each one a 2-loop. Next, we attach a $(5s+1)$-loop to b^q_{2s} and a $(5s+2)$-loop to b^q_{2s+1} with D_q-use s. However, throughout the construction, we may change the position of loops or add new loops to specific components of $\mathcal{B}_q$ depending on enumerations into A_q (and thus into D_q). For the $2s$th and $(2s+1)$st components of $\mathcal{B}_q$, we have three possible final configurations of the loops. If we never start the process of diagonalizing using these components, then they will remain the same forever:

$$b^q_{2s} : 2, 5s+1$$
$$b^q_{2s+1} : 2, 5s+2.$$

If we start, but don't finish, diagonalizing using these components, they will end in the following configuration:

$$b^q_{2s} : 2, 5s+1, 5s+3$$
$$b^q_{2s+1} : 2, 5s+1, 5s+4$$

If we complete a diagonalization with these components, then they will end as:

$$b^q_{2s} : 2, 5s+1, 5s+2, 5s+4$$
$$b^q_{2s+1} : 2, 5s+1, 5s+2, 5s+3.$$

For all $e \in \omega$, we meet the requirement

$$R^q_e : \Phi_e^{D_q} : \mathcal{G} \to \mathcal{B}_q \text{ is not an isomorphism.}$$

To satisfy this requirement, we will diagonalize against $\Phi_e^{D_q}$. Let α be an R^q_e-strategy.

When α is first eligible to act, it picks a large number n_α, and for the rest of this strategy, let $n = n_\alpha$. This parameter indicates which connected components of $\mathcal{B}_q$ will be used in the diagonalization. At future stages, α checks if $\Phi_e^{D_q}$ maps the $2n$th and $(2n+1)$st connected component of $\mathcal{G}$ to the $2n$th and $(2n+1)$st connected component of $\mathcal{B}_q$, respectively. If not, α does not take any action. If α sees such a computation, it defines m_α to be the max of the uses of the computations on each component and restrains $D_q \upharpoonright m_\alpha + 1$.

At this point, our connected components in $\mathcal{G}[s]$ and $\mathcal{B}_q[s]$ are as follows:

$$a_{2n} : 2, 5n+1 \quad b^q_{2n} : 2, 5n+1$$
$$a_{2n+1} : 2, 5n+2 \quad b^q_{2n+1} : 2, 5n+2.$$

Since $\Phi_e^{D_q}$ looks like a potential isomorphism between $\mathcal{G}$ and $\mathcal{B}_q$, α will now take action to eventually force the true isomorphism to match a_{2n} with b^q_{2n+1} and to match a_{2n+1} with b^q_{2n} while preventing $\Phi_e^{D_q}$ from correcting itself on these components. Furthermore, α must do this in a way that will allow other requirements to succeed.

After m_α has been defined, α adds a $(5n+3)$-loop to a_{2n} and a $(5n+4)$-loop to a_{2n+1} in $\mathcal{G}[s]$. It also attaches a $(5n+3)$-loop to b^q_{2n} and a $(5n+4)$-loop to b^q_{2n+1} in $\mathcal{B}_q$[s]. Let v_α be a large unused number and set the use of all edges in these new loops appearing in $\mathcal{B}_q[s]$ to be $\langle v_\alpha, q\rangle$. Notice that $\langle v_\alpha, q\rangle > m_\alpha$ and that enumerating v_α into A_q will put $\langle v_\alpha, q\rangle$ into D_q, removing the $(5n+3)$- and $(5n+4)$-loops from $\mathcal{B}_q$ but not the $(5n+1)$- or $(5n+2)$-loops. Our connected components in $\mathcal{G}[s]$ and in $\mathcal{B}_q[s]$ are now:

$$a_{2n} : 2, 5n+1, 5n+3 \quad b^q_{2n} : 2, 5n+1, 5n+3$$
$$a_{2n+1} : 2, 5n+2, 5n+4 \quad b^q_{2n+1} : 2, 5n+2, 5n+4.$$

After adding the $(5n+3)$- and $(5n+4)$-loops to both graphs, α must now wait for higher priority strategies which have already defined their maps on the $2m$th and $(2m+1)$st components for all $m \leq n$ to recover their maps before taking the last step. If β is a higher priority S or T strategy for which f_β or $g_\beta^{D_p}$ is already defined on the $2n$th and $(2n+1)$st components of $\mathcal{G}$, then before completing its diagonalization, α must wait for β to extend f_β

or $g_\beta^{D_p}$ to be defined on the new $(5n+3)$- and $(5n+4)$-loops. We will refer to this action as α issuing a challenge to all higher priority S and T requirements.

Once all higher priority strategies recover, α enumerates v_α into A_q (and thus $\langle v_\alpha, q\rangle$ goes into D_q). Doing this causes the $(5n+3)$-loop attached to b^q_{2n} and the $(5n+4)$-loop attached to b^q_{2n+1} to disappear in $\mathcal{B}_q[s]$. We now attach a $(5n+4)$-loop to b^q_{2n} and a $(5n+3)$-loop to b^q_{2n+1}. We also attach a $(5n+1)$-loop to a_{2n+1} and b^q_{2n+1} and a $(5n+2)$-loop to a_{2n} and b^q_{2n}, and we will refer to this process as homogenizing the components in $\mathcal{G}$ and in $\mathcal{B}_q$. The final configuration of our loops is:

$$\begin{array}{rlrl} a_{2n} : & 2, 5n+1, 5n+2, 5n+3 & \quad b^q_{2n} : & 2, 5n+1, 5n+2, 5n+4 \\ a_{2n+1} : & 2, 5n+1, 5n+2, 5n+4 & \quad b^q_{2n+1} : & 2, 5n+1, 5n+2, 5n+3. \end{array}$$

By homogenizing the components, we ensured that when we added loops for the diagonalization in $\mathcal{B}_q$, we also made adjustments in $\mathcal{G}$ to keep the components isomorphic to each other. Additionally, because $v_\alpha > m_\alpha$, the values $\Phi_e^{D_q}(a_{2n}) = b_{2n}$ and $\Phi_e^{D_q}(a_{2n+1}) = b_{2n+1}$ remain. So if $\Phi_e^{D_q}[s]$ is extended to a map on the entirety of $\mathcal{G}$, it cannot be a D_q-computable isomorphism, and so R^q_e is satisfied. If we meet R^q_e for all $e \in \omega$, we have that $\mathcal{G}$ is not computably categorical relative to D_q with $\mathcal{B}_q$ being the witness.

1.2.5 Interactions between multiple strategies

There are some interactions which can cause problems between these strategies. We will explain how the strategies described in this section, with some tweaks, can solve these issues.

We first point out that the independence requirements cause no serious issues for the other requirements. An N^p_e-strategy α, when it is first eligible to act, will pick a large unused number x_α, and so if it ever enumerates x_α into A_p, it will not violate any restraints placed by higher priority independence or R^q_e requirements. If this enumeration injures loops in or embeddings defined on components of $\mathcal{M}_i^{D_r}$ for some higher priority T^r_i-strategy where $p \leq r$, the T^r_i-strategy will be able to check for this D_r change when it is next eligible to act and will be able to react accordingly to succeed.

The next main interaction to note is between an R_e^q-strategy β and an S- or T-strategy α where $\alpha^\frown\langle\infty\rangle \subseteq \beta$. In the tree of strategies, $\alpha^\frown\langle\infty\rangle \subseteq \beta$ indicates that β guesses α will define an embedding of $\mathcal{G}$ into its graph $\mathcal{M}_i$ or $\mathcal{M}_i^{D_p}$. In the informal description for β, we had β wait for higher priority strategies to recover their embeddings defined on $\mathcal{G}$ after we added $(5n+3)$- and $(5n+4)$-loops to components of $\mathcal{G}$. When β adds the new loops, it updates α's parameter by setting $n_\alpha = n_\beta$ if α is an S-strategy. If α is instead a T-strategy, then β updates n_α to be the least $m \leq n_\beta$ such that $g_\alpha^{D_p}$ is no longer fully defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$ (for reasons that will become clear below). In either case, this causes α to return to previous components of $\mathcal{G}$ to find copies of them in either $\mathcal{M}_e$ or $\mathcal{M}_i^{D_p}$. If it is the case that either $\mathcal{G} \cong \mathcal{M}_e$ or $\mathcal{G} \cong \mathcal{M}_i^{D_p}$, α will eventually find copies and is able to define either a computable or D_p-computable isomorphism between the two graphs. Hence, the only tweaks needed for the S_e- and T_i^p-strategies α are steps in which they check if there is a lower priority R_e^q-strategy β where $\alpha^\frown\langle\infty\rangle \subseteq \beta$ that has issued its challenge after adding the new initial loops in $\mathcal{G}$.

There is a related technical point concerning R_e^q homogenizing the $2n_\beta$th and $(2n_\beta+1)$st components in the last step of its diagonalization. We do not ask the higher priority S- and T-strategies α with $\alpha^\frown\langle\infty\rangle \subseteq \beta$ to go back and match these final homogenizing loops. Instead, we will extend f_α (or $g_\alpha^{D_p}$) to those homogenizing loops in a computable (or D_p-computable) way for α on the true path in the verification.

One last interaction arises between an R_e^q-strategy β and a T_i^p-strategy α where $\alpha^\frown\langle\infty\rangle \subseteq \beta$ and $q < p$ in P. With the current construction, we could have the following situation which makes it impossible for α to succeed. Suppose α finds copies at a stage s_0 of the $2n_\beta$th and $(2n_\beta+1)$st components of $\mathcal{G}[s_0]$ into $\mathcal{M}_i^{D_p}[s_0]$, and we have the following components in both graphs

$$a_{2n_\beta} : 2, 5n_\beta + 1 \quad c : 2, 5n_\beta + 1$$

$$a_{2n_\beta+1} : 2, 5n_\beta + 2 \quad d : 2, 5n_\beta + 2$$

where c and d are the root nodes of the copies of the $\mathcal{G}$ components found in $\mathcal{M}_i^{D_p}[s_0]$. At this point, α defines $g_\alpha^{D_p}[s_0]$ by mapping $a_{2n_\beta} \mapsto c$ and $a_{2n_\beta+1} \mapsto d$ with a use $\langle u_{\alpha,n_\beta}, p\rangle$.

Suppose at a later stage $s_1 > s_0$, β adds new loops to the corresponding components in $\mathcal{G}$ with a D_q-use of $\langle v_\alpha, q\rangle$ for v_α large and issues its challenge, and so our components in $\mathcal{G}[s_1]$ and $\mathcal{M}_i^{D_p}[s_1]$ are

$$a_{2n_\beta} : 2, 5n_\beta + 1, 5n_\beta + 3 \quad c : 2, 5n_\beta + 1$$

$$a_{2n_\beta+1} : 2, 5n_\beta + 2, 5n_\beta + 4 \quad d : 2, 5n_\beta + 2.$$

Then, suppose at stage $s_2 > s_1$ that $\Phi_i^{D_p}[s_1]$ adds the new loops correctly to c and d:

$$a_{2n_\beta} : 2, 5n_\beta + 1, 5n_\beta + 3 \quad c : 2, 5n_\beta + 1, 5n_\beta + 3$$

$$a_{2n_\beta+1} : 2, 5n_\beta + 2, 5n_\beta + 4 \quad d : 2, 5n_\beta + 2, 5n_\beta + 4.$$

Let z_{α,n_β} be the minimum use for any of the new edges in these loops, and assume that $\langle v_\alpha, q\rangle < z_{\alpha,n_\beta}$. The strategy α extends $g_\alpha^{D_p}[s_1]$ to map the $(5n_\beta + 3)$- and $(5n_\beta + 4)$-loops from $\mathcal{G}$ into $\mathcal{M}_i^{D_p}$ with a large use (i.e., greater than $\langle u_{\alpha,n_\beta}, p\rangle$). α has now met its challenge and takes the ∞ outcome.

Finally, suppose at a stage $s_3 > s_2$, β is eligible to act again. β enumerates v_α into A_q, which enumerates $\langle v_\alpha, q\rangle$ into D_q and D_p since $q < p$. Since $\langle v_\alpha, q\rangle \in D_q$, the $(5n_\beta + 3)$- and $(5n_\beta + 4)$-loops in $\mathcal{B}_q$ disappear, and so β can homogenize the $2n_\beta$th and $(2n_\beta + 1)$st components in $\mathcal{G}$ and in $\mathcal{B}_q$ and diagonalize.

$$a_{2n_\beta} : 2, 5n_\beta + 1, 5n_\beta + 2, 5n_\beta + 3 \quad b^q_{2n_\beta} : 2, 5n_\beta + 1, 5n_\beta + 2, 5n_\beta + 4$$

$$a_{2n_\beta+1} : 2, 5n_\beta + 1, 5n_\beta + 2, 5n_\beta + 4 \quad b^q_{2n_\beta+1} : 2, 5n_\beta + 1, 5n_\beta + 2, 5n_\beta + 3.$$

The map $\Phi_e^{D_q}[s_3]$ still maps a_{2n_β} to $b^q_{2n_\beta}$ and $a_{2n_\beta+1}$ to $b^q_{2n_\beta+1}$ since the D_q-use for these computations is less than v_α and thus less than $\langle v_\alpha, q\rangle$.

However, because $\langle v_\alpha, q\rangle$ has gone into D_p as well and $\langle v_\alpha, q\rangle < z_{\alpha,n_\beta}$, the $(5n_\beta+3)$- and $(5n_\beta+4)$-loops in $\mathcal{M}_i^{D_p}[s_3]$ also disappear. But v_α was chosen to be large at stage s_1, so $v_\alpha > u_{\alpha,n_\beta}$ and so $g_\alpha^{D_p}[s_3]$ still maps a_{2n_β} to c and $a_{2n_\beta+1}$ to d. This allows the opponent controlling $\mathcal{M}_i^{D_q}$ to add loops in the following way to diagonalize:

$$\begin{aligned} a_{2n_\beta} &: 2, 5n_\beta+1, 5n_\beta+2, 5n_\beta+3 & c &: 2, 5n_\beta+1, 5n_\beta+2, 5n_\beta+4 \\ a_{2n_\beta+1} &: 2, 5n_\beta+1, 5n_\beta+2, 5n_\beta+4 & d &: 2, 5n_\beta+1, 5n_\beta+2, 5n_\beta+3. \end{aligned}$$

This now makes it impossible for α to succeed if $\mathcal{G} \cong \mathcal{M}_i^{D_p}$.

To solve this conflict, α needs to lift the use of $g_\alpha^{D_p}[s_1]$ when β starts its diagonalization process. Specifically, when β adds the $(5n_\beta+3)$- and $(5n_\beta+4)$-loops to $\mathcal{B}_q$ and sets their D_q-use to be $\langle v_\alpha, q\rangle$, we enumerate u_{α,n_β} into A_p, which puts $\langle u_{\alpha,n_\beta}, p\rangle$ into D_p but nothing into D_q. This action makes $g_\alpha^{D_p}$ undefined on the $2n_\beta$th and $(2n_\beta+1)$st components of $\mathcal{G}$. When α is next eligible to act, it will redefine $g_\alpha^{D_p}$ on the $2n_\beta$th and $(2n_\beta+1)$st components of $\mathcal{G}$ with a large use greater than $\langle v_\alpha, q\rangle$. Therefore, if β later enumerates v_α into A_q to diagonalize, the map $g_\alpha^{D_p}$ will become undefined on the entirety of the $2n_\beta$th and $(2n_\beta+1)$st components, preventing the opponent from using $\mathcal{M}_i^{D_p}$ to diagonalize against α.

It is possible that there is more than one T-strategy α with $\alpha^\frown\langle\infty\rangle \subseteq \beta$ associated with elements in P greater than q. In this case, β has to enumerate the use u_{α,n_β} for each such α into A_q. These elements may cause $g_\alpha^{D_p}$ to become undefined on the $2m$th and $(2m+1)$st components for $m < n_\beta$, or for these components in $\mathcal{M}_\alpha^{D_p}$ to disappear. Therefore, for a T_i^p-strategy α with $\alpha^\frown\langle\infty\rangle \subseteq \beta$ and $q < p$, β resets n_α to be the least $m \leq n_\beta$ such that $g_\alpha^{D_p}$ no longer matches the $2m$th and $(2m+1)$st components in $\mathcal{G}$ and $\mathcal{M}_i^{D_p}$.

1.3 Proof of Theorem 1.1.10

In this section, we prove Theorem 1.1.10. Fix a computable partially ordered set $P = (P, \leq)$, and let $P = P_0 \sqcup P_1$ be a computable partition of P.

We build our computable directed graph $\mathcal{G}$ stage by stage as outlined in section 1.2.2, and for each $q \in P_1$, we build an isomorphic copy $\mathcal{B}_q$ of $\mathcal{G}$ as outlined in section 1.2.4.

We will also build a uniformly c.e. family of independent sets A_p for $p \in P$ via a priority argument on a tree of strategies. We define

$$D_p = \bigoplus_{q \leq p} A_q$$

and

$$\overline{D_p} = \bigoplus_{q \neq p} A_q.$$

Our embedding h of P into the c.e. degrees is the map $h(p) = D_p$ for all $p \in P$.

1.3.1 Requirements

Recall our four types of requirements for our construction:

$$N_e^p : \Phi_e^{\overline{D_p}} \neq A_p$$

$$S_e : \text{if } \mathcal{G} \cong \mathcal{M}_e, \text{ then there exists a computable isomorphism } f_e : \mathcal{G} \to \mathcal{M}_e$$

$$T_i^p : \text{if } \mathcal{G} \cong \mathcal{M}_i^{D_p}, \text{ then there exists a } D_p\text{-computable isomorphism } g_i^{D_p} : \mathcal{G} \to \mathcal{M}_i^{D_p}$$

$$R_e^q : \Phi_e^{D_q} : \mathcal{G} \to \mathcal{B}_q \text{ is not an isomorphism}$$

1.3.2 Construction

Let $\Lambda = \{\infty <_\Lambda \cdots <_\Lambda w_2 <_\Lambda s <_\Lambda w_1 <_\Lambda w_0\}$ be the set of outcomes, and let $T = \Lambda^{<\omega}$ be our tree of strategies. The construction will be performed in ω many stages s.

We define the **current true path** p_s, the longest strategy eligible to act at stage s, inductively. For every s, λ, the empty string, is eligible to act at stage s. Suppose the strategy α is eligible to act at stage s. If $|\alpha| < s$, then follow the action of α to choose a successor $\alpha^\frown\langle o\rangle$ on the current true path. If $|\alpha| = s$, then set $p_s = \alpha$. For all strategies β such that $p_s <_L \beta$, initialize β (i.e., set all parameters associated to β to be undefined). If $\beta <_L p_s$ and $|\beta| < s$, then β retains the same values for its parameters.

We will now give formal descriptions of each strategy and their outcomes in the construction.

1.3.3 N_e^p-strategies and outcomes

We first cover the N_e^p-strategies used to make each c.e. set A_p independent. Let α be an N_e^p-strategy eligible to act at stage s.

Case 1: If α is acting for the first time at stage s or has been initialized since the last α-stage, define its parameter x_α to be large, and take outcome w_0.

Case 2: If x_α is already defined and α took outcome w_0 at the last α-stage, check if

$$\Phi_e^{\overline{D_p}}(x_\alpha)[s]\downarrow= 0.$$

If not, take the w_0 outcome. If $\Phi_e^{\overline{D_p}}(x_\alpha)[s]\downarrow= 0$, enumerate x_α into A_p and take the s outcome which will preserve $\overline{D_p} \upharpoonright (\text{use}(\Phi_e^{\overline{D_p}}(x_\alpha)[s]) + 1)$.

Case 3: If α took the s outcome the last time it was eligible to act and has not been initialized, take the s outcome again.

1.3.4 S_e-strategies and outcomes

We now detail our S_e-strategy to make $\mathcal{G}$ computably categorical. Let α be an S_e-strategy eligible to act at stage s.

Case 1: If α is acting for the first time or has been initialized since the last α-stage, define $n_\alpha = 0$ and $f_\alpha[s]$ to be the empty map. Take the w_0 outcome.

Case 2: If α has defined n_α and is currently challenged by an R_e^q-strategy β with

$\alpha^\frown\langle\infty\rangle \subseteq \beta$, then α acts as follows. In the verification, we show that there can only be one strategy challenging α at a time. When β challenged α, if the value of n_α was greater than n_β, then β redefined n_α to be equal to n_β. Therefore, we currently have $n_\alpha \leq n_\beta$. Furthermore, if $n_\alpha = n_\beta$, then f_α may already be defined on the $(5n_\alpha + 1)$- and $(5n_\alpha + 2)$-loops in the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{G}$.

α searches for copies of the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{G}$ in $\mathcal{M}_e[s]$. More formally, if f_α is not defined on any loops in these components, then α searches for full copies of both components in $\mathcal{M}_e[s]$. If $n_\alpha = n_\beta$ and f_α is already defined on the $(5n_\alpha + 1)$- and $(5n_\alpha + 2)$-loops in $\mathcal{G}$, then α searches for the new loops of lengths $5n_\alpha + 3$ and $5n_\alpha + 4$ in the matched components in $\mathcal{M}_e[s]$. If no copies are found, set $f_\alpha[s] = f_\alpha[s-1]$, leave n_α unchanged, and take the w_{n_α} outcome. If copies are found, extend $f_\alpha[s-1]$ to $f_\alpha[s]$ by matching the components, increment n_α by 1 and check if $n_\alpha > n_\beta$ for this new n_α. If yes, take the ∞ outcome and declare β's challenge to have been met. If not, take the w_{n_α} outcome and let β's challenge remain active.

Case 3: If α has defined its parameter n_α and α is not currently challenged, then, α continues to search for copies of the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{G}$ in $\mathcal{M}_e[s]$. If no copies are found, define $f_\alpha[s] = f_\alpha[s-1]$, leave n_α unchanged, and take the w_{n_α} outcome. Otherwise, extend $f_\alpha[s-1]$ to $f_\alpha[s]$ by mapping the components to their respective copies in $\mathcal{M}_e[s]$, increment n_α by 1, and take the ∞ outcome.

1.3.5 T_i^p-strategies and outcomes

For each $p \in P_0$, we have the following T_i^p-strategy. Let α be a T_i^p-strategy eligible to act at stage s.

Case 1: If α is acting for the first time or has been initialized since the last α-stage, set $n_\alpha = 0$, define $g_\alpha^{D_p}[s]$ to be the empty function, and take the w_0 outcome.

Case 2: α is currently challenged by an R_e^q-strategy β where $\alpha^\frown\langle\infty\rangle \subseteq \beta$. Let s_0 be the stage at which β challenged α. In the verification, we will show that there can only be one strategy challenging α at a time. When β challenged α at stage s_0, it redefined n_α to equal the least $m \leq n_\beta$ such that $g_\alpha^{D_p}$ is not fully defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$.

If $q < p$ and s is the first α-stage since s_0 and n_α was greater than n_β at stage s_0, then we have to perform a preliminary action. In this case, β enumerated u_{α,n_β} into A_p, causing the map $g_\alpha^{D_p}[s_0]$ to become undefined on the $2n_\beta$th and $(2n_\beta+1)$st components of $\mathcal{G}$. Choose a new large number u_{α,n_β} and redefine $g_\alpha^{D_p}[s]$ to be equal to $g_\alpha^{D_p}[s_0]$ on the 2-loops, $(5n_\beta+1)$-loops, and $(5n_\beta+2)$-loops in these components with use $\langle u_{\alpha,n_\beta}, p\rangle$. This ends the preliminary step.

Next, we perform the main action in this case. If $n_\alpha = n_\beta$ and $g_\alpha^{D_p}$ is already defined on the $(5n_\alpha+1)$- and $(5n_\alpha+2)$-loops of the $2n_\alpha$th and $(2n_\alpha+1)$st components in $\mathcal{G}$, then α searches for the oldest and lexicographically least copies of the $(5n_\alpha+3)$- and $(5n_\alpha+4)$-loops in $\mathcal{M}_i^{D_p}[s]$. If $g_\alpha^{D_p}$ is not currently defined on any of the loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$, then α searches for the oldest and lexicographically least copies of these components in $\mathcal{M}_i^{D_p}[s]$. In either case, if such copies are found, extend $g_\alpha^{D_p}[s]$ to map onto these copies with use $\langle u_{\alpha,n_\alpha}, p\rangle$ for large u_{α,n_α}, increment n_α by 1, and check if $n_\alpha > n_\beta$ for this new n_α. If yes, take the ∞ outcome and declare β's challenge to α to be met. If not, then take the w_{n_α} outcome and let β's challenge to α remain active.

Case 3: α is not currently challenged by an R_e^q-strategy. Let t be the last α-stage. In this case, α defined $g_\alpha^{D_p}[t]$ on the $2m$th and $(2m+1)$st components with use $\langle u_{\alpha,m}, p\rangle$ for $m < n_\alpha$. Let l_m be the max D_p-use for the computation of a loop in the image of the $2m$th and $(2m+1)$st components under $g_\alpha^{D_p}[t]$. In the verification, we will show that $l_m < u_{\alpha,m}$ for all $m < n_\alpha$.

Step 1: If there is an $m < n_\alpha$ such that $D_p[t] \upharpoonright l_m \neq D_p[s] \upharpoonright l_m$, then let m be the least such value. Note that for $m \leq m^* < n_\alpha$, the map $g_\alpha^{D_p}$ is now undefined on the $2m^*$th and $(2m^*+1)$st components of $\mathcal{G}$. The loops in the image of the $2k$th and $(2k+1)$st components of $\mathcal{G}$ under $g_\alpha^{D_p}[t]$ for $k < m$ remain in $\mathcal{M}_i^{D_p}[s]$. Update $n_\alpha = m$.

Step 2: By the update in Step 1, we have that for each $m < n_\alpha$ that $D_p[t] \upharpoonright l_m = D_p[s] \upharpoonright l_m$. For each $m < n_\alpha$, if any, where $D_p[t] \upharpoonright \langle u_{\alpha,m}, p\rangle \neq D_p[s] \upharpoonright \langle u_{\alpha,m}, p\rangle$, set $g_\alpha^{D_p}[s] = g_\alpha^{D_p}[t]$ on the loops in $\mathcal{G}$ in the $2m$th and $(2m+1)$st components with the same use as at stage t.

Step 3: We can now perform the main action of this case. α searches for the oldest and lexicographically least copies of the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$ in $\mathcal{M}_i^{D_p}[s]$. If no copies are found, leave $g_\alpha^{D_p}$ and n_α unchanged and take outcome w_{n_α}. Otherwise, extend $g_\alpha^{D_p}$ by mapping the loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$ to their copies in $\mathcal{M}_i^{D_p}$ with use $\langle u_{\alpha,n_\alpha}, p\rangle$ where u_{α,n_α} is chosen large, increment n_α by 1, and take the ∞ outcome.

1.3.6 R_e^q-strategies and outcomes

Finally, for each $q \in P_1$, we have the following R_e^q-strategy. Let α be an R_e^q-strategy eligible to act at stage s.

Case 1: If α is first eligible to act at stage s or has been initialized, define the parameter $n_\alpha = n$ to be large and take outcome w_0.

Case 2: If we are not in **Case 1** and α took outcome w_0 at the last α-stage, check whether $\Phi_e^{D_q}[s]$ maps the $2n$th and $(2n+1)$st components of $\mathcal{G}$ isomorphically into $\mathcal{B}_q$. If not, take outcome w_0.

If so, set m_α to be the maximum D_q-use of these computations. Let v_α be large. Add a $(5n+3)$-loop to a_{2n} in $\mathcal{G}$ (computably) and to b_{2n}^q in $\mathcal{B}_q$ (with D_q-use $\langle v_\alpha, q\rangle$) and add a $(5n+4)$-loop to a_{2n+1} in $\mathcal{G}$ (computably) and to b_{2n+1}^q in $\mathcal{B}_q$ (with D_q-use $\langle v_\alpha, q\rangle$).

For each T_i^p-strategy γ where $\gamma^\frown\langle\infty\rangle \subseteq \alpha$ and $q < p$, enumerate the use $u_{\gamma,n}$ into A_p (and so $\langle u_{\gamma,n}, p\rangle$ enters D_p), and challenge γ. Note that if $n_\gamma < n_\alpha$, then there is no $u_{\gamma,n}$ to enumerate into A_p. For each S-strategy β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, challenge β and reset $n_\alpha = n_\beta$ if $n_\alpha > n_\beta$. Otherwise, leave n_β as it is. For each T-strategy β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, reset n_β to be the least $m \leq n_\alpha$ such that $g_\alpha^{D_p}$ does not match all of the $2m$th and $(2m+1)$st components of $\mathcal{G}$. Take outcome w_1.

Case 3: If α took outcome w_1 at the last α-stage, then enumerate v_α into A_q, move the

$(5n+3)$-loop in $\mathcal{B}_q$ from b^q_{2n} to b^q_{2n+1}, and move the $(5n+4)$-loop in from b^q_{2n+1} to b^q_{2n}. Attach a $(5n+1)$-loop to a_{2n+1} and b^q_{2n+1} and a $(5n+2)$-loop to a_{2n} and b^q_{2n} and define the D_q-use of these new loops to be large. Take outcome s.

Case 4: If α took outcome s at the last α-stage and has not been initialized, then take outcome s.

1.3.7 Verification

We first prove that the map h where $h(p) = D_p$ is an embedding into the c.e. degrees (if all N^p_e requirements are satisfied) and the computable and D_p-computable embeddings for $p \in P_0$, if they are defined, are the isomorphisms needed for $\mathcal{G}$'s categoricity. We will then prove key observations about the construction before stating the main verification lemma.

Lemma 1.3.1. Suppose that for all $p \in P$ and $e \in \omega$, the requirement N^p_e is satisfied. Then, for all $p, q \in P$, we have that $p \leq q$ if and only if $D_p \leq_T D_q$.

Proof. Assume that each N^p_e requirement was satisfied and suppose that $p \leq q$. Since $p \leq q$, for all r where $r \leq p$, we have that $r \leq q$ as well and so $D_p \leq_T D_q$. If $p \not\leq q$, then since

$$A_p \not\leq_T \bigoplus_{t \neq p} A_t,$$

it immediately follows that $A_p \not\leq_T \bigoplus_{t \leq q} A_t$ and hence $D_p \not\leq_T D_q$. □

Lemma 1.3.2. If $f : \mathcal{G} \to \mathcal{G}$ is an embedding of $\mathcal{G}$ into itself, then f is an isomorphism (and is, in fact, the identity map).

Proof. Let $f : \mathcal{G} \to \mathcal{G}$ be an embedding. Since embeddings preserve loops and only the root nodes a_m are contained in more than one loop, f must map root nodes to root nodes. Furthermore, since only the root nodes a_{2n} and a_{2n+1} can have $(5n+1)$-loops, we can only have that $f(a_{2n}) = a_{2n}$ or $f(a_{2n}) = a_{2n+1}$. However, the only situation in which a_{2n+1} has a $(5n+1)$-loop is when we attached a $(5n+3)$-loop to a_{2n} but *not* to a_{2n+1}. Thus, $f(a_{2n}) = a_{2n}$

and similarly, $f(a_{2n+1}) = a_{2n+1}$. Since $\mathcal{G}$ is a directed graph, it must map the loops attached to a_{2n} and to a_{2n+1} identically onto themselves. $\square$

Lemma 1.3.3. If $\mathcal{M}_e \cong \mathcal{G}$ for a computable directed graph $\mathcal{M}_e$ and $f_e : \mathcal{G} \to \mathcal{M}_e$ is an embedding defined on all of $\mathcal{G}$, then f_e is an isomorphism.

Proof. This follows immediately from Lemma 1.3.2. $\square$

By Lemma 1.3.3, if $\mathcal{M}_e \cong \mathcal{G}$ or $\mathcal{G} \cong \mathcal{M}_i^{D_p}$, then there is a unique isomorphism from $\mathcal{G}$ to $\mathcal{M}_e$ and from $\mathcal{G}$ to $\mathcal{M}_i^{D_p}$. We refer to the image of the $2n$th and $(2n+1)$st components of $\mathcal{G}$ in $\mathcal{M}_e^{D_p}$ as the **true copies** of these components in $\mathcal{M}_i^{D_p}$. We now prove several auxiliary lemmas about the construction.

Lemma 1.3.4. If $\mathcal{G} \cong \mathcal{M}_i^{D_p}$, then for each n, there is an s such that for all $t \geq s$, the true copies of the $2n$th and $(2n+1)$st components of $\mathcal{G}$ in $\mathcal{M}_i^{D_p}$ are the oldest and lexicographically least isomorphic copies in $\mathcal{M}_i^{D_p}[t]$ of these components.

Proof. Let u be the maximum D_p-use for the edges in the true copies of these components in $\mathcal{M}_i^{D_p}$ and let s_0 be such that $D_p \upharpoonright (u+1)[s_0] = D_p \upharpoonright (u+1)$. Because D_p is c.e., the true components will be defined at every stage $s \geq s_0$. There may be a finite number of older fake copies of these components, but they will disappear as numbers enter D_p, and so for a large enough $s \geq s_0$, the true copies will be the oldest and lexicographically least in $\mathcal{M}_i^{D_p}[s]$. $\square$

Lemma 1.3.5. Let α be an N_e^p-strategy that enumerates x_α into A_p at stage s. Unless α is initialized, no number below use$(\Phi_e^{\overline{D_p}}(x_\alpha)[s])$ is enumerated into $\overline{D_p}$ after stage s.

Proof. After α enumerates x_α into A_p at stage s, all strategies extending $\alpha^\frown\langle s\rangle$ will define new large parameters greater than use$(\Phi_e^{\overline{D_p}}(x_\alpha)[s])$. The only strategies which have parameters smaller than use$(\Phi_e^{\overline{D_p}}(x_\alpha)[s])$ are to the left of α on the tree of strategies or are R- or N-strategies β such that $\beta \subset \alpha$. When β enumerates a number into their assigned c.e. set, then it will take outcome s and initialize α. $\square$

Lemma 1.3.6. An S_e-strategy or a T_i^p-strategy can be challenged by at most one R-strategy at any given stage.

Proof. Let α be an S_e-strategy (or T_i^p-strategy) and suppose that there exists some β such that $\alpha^\frown\langle\infty\rangle \subseteq \beta$ and β is an R-strategy that challenges α. If β challenges α at a stage s, then β takes the w_1 outcome for the first time. The strategies extending $\beta^\frown\langle w_1\rangle$ will choose witnesses at stage s, and in particular, none will challenge α. So, at most one strategy will challenge α at stage s. Until α is able to match the newly added loops in $\mathcal{G}$ to meet the challenge, α will take outcome w_{n_α}. R-strategies γ such that $\gamma \supseteq \alpha^\frown\langle w_{n_\alpha}\rangle$ will not challenge α since $w_{n_\alpha} \neq \infty$. □

Lemma 1.3.7. Suppose α is an R_e^q-strategy that is never initialized after stage s. Then α can only challenge higher priority S-strategies and T-strategies at most once after stage s.

Proof. Suppose α is an R_e^q-strategy that is never initialized after stage s and suppose it challenges all S-strategies and T-strategies β such that $\beta^\frown\langle\infty\rangle \subseteq \alpha$. Because α is never initialized again, if we ever return to α, it will take the s outcome as it can now diagonalize, and will continue to take the s outcome at all subsequent α-stages. If we do not return to α, α will not be able to challenge any higher priority S-strategies or T-strategies after stage s since it will never be eligible to act again. □

Lemma 1.3.8. At most one strategy α enumerates numbers at any stage.

Proof. Suppose numbers are enumerated at a stage s and α is the highest priority strategy which enumerates a number. α must either be for an R_e^q or an N_e^p requirement. If α is an N_e^p-strategy, it will take the s outcome for the first time, and if α is an R_e^q-strategy, it will either take the w_1 outcome or the s outcome for the first time. In either case, the remaining strategies which act at stage s will act by simply defining their parameters and taking the w_0 outcome. Therefore, α is the only strategy to enumerate a number at stage s. □

Lemma 1.3.9. Let α be a T_i^p-strategy that defines $g_\alpha^{D_p}[s_m]$ on the $2m$th and $(2m+1)$st components of $\mathcal{G}$ at stage s_m. Until α is initialized (if ever), only strategies β such that $\alpha^\frown\langle\infty\rangle \subseteq \beta$ can enumerate a number $n \leq u_{\alpha,m}$ at any stage $t \geq s_m$.

Proof. Let α be such a T_i^p-strategy. If α is not challenged, after defining $g_\alpha^{D_p}[s_m]$ on the $2m$th and $(2m+1)$st components of $\mathcal{G}$ at stage s_m, it takes the ∞ outcome. Hence, all strategies extending $\alpha^\frown\langle w_k\rangle$ for some k or to the right of α are initialized. These strategies will choose new parameters larger than $u_{\alpha,m}$ when they are next eligible to act. They are also the only strategies not extending $\alpha^\frown\langle\infty\rangle$ which can enumerate numbers without initializing α, so it suffices to show that they will not enumerate numbers below $u_{\alpha,m}$. Let β be such a strategy.

If β is an $N_{e'}^{p'}$-strategy, then it can only enumerate its parameter x_β into $A_{p'}$, and it was chosen such that $x_\beta > u_{\alpha,m}$. If β is an R_e^q-strategy then it enumerates two types of numbers: v_β and u_{γ,n_β} for any $T_{i'}^{p'}$-strategy such that $\gamma^\frown\langle\infty\rangle \subseteq \beta$ and $p' < q$. Since v_β will be chosen large after stage s_m, we have that $v_\beta > u_{\alpha,m}$. Since $\beta \supseteq \alpha^\frown\langle w_m\rangle$, we have that $n_\beta > u_{\alpha,m}$ because n_β was chosen to be large after $u_{\alpha,m}$ was defined. In particular, when n_β is chosen, γ cannot have defined u_{γ,n_β} since n_β is large, so when γ defines u_{γ,n_β} later, it must satisfy $u_{\gamma,n_\beta} > n_\beta$. Hence $u_{\gamma,n_\beta} > u_{\alpha,m}$.

In the case in which α defines $g_\alpha^{D_p}[s_m]$ on the mentioned components while it is being challenged, then strategies β which extend $\alpha^\frown\langle w_k\rangle$ have been initialized before when α took the ∞ outcome at some previous stage. So, when β acts again, it defines new parameters larger than $u_{\alpha,m}$, and the rest of the argument is largely the same as above. $\square$

Lemma 1.3.10. Let α be a T_i^p-strategy that takes a w_k outcome at a stage s. Let t be the next α-stage. Unless α has been initialized, $D_p[t] \upharpoonright \langle u_{\alpha,m}, p\rangle = D_p[s] \upharpoonright \langle u_{\alpha,m}, p\rangle$ for all $m < n_\alpha$, and so $g_\alpha^{D_p}[t]$ remains defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$ for all $m < n_\alpha$.

Proof. This follows immediately from Lemma 1.3.9. $\square$

Lemma 1.3.11. Let α be a T_i^p-strategy. If α defines $g_\alpha^{D_p}$ on the $2m$th and $(2m+1)$st components of $\mathcal{G}$ at stage s, then $l_m[s] < u_{\alpha,m}[s]$ where $l_m[s]$ is the max use of the edges in the $(5m+1)$- and $(5m+2)$-loops in the copies of the $2m$th and $(2m+1)$st components of $\mathcal{G}$ in $\mathcal{M}_i^{D_p}$ and $\langle u_{\alpha,m}[s], p\rangle$ is the D_p-use of $g_\alpha^{D_p}[s]$ on these components. Furthermore, for all α-stages $t > s$, we have $u_{\alpha,m}[t] \geq u_{\alpha,m}[s] > l_m[s] = l_m[t]$ unless these components in $\mathcal{M}_i^{D_p}$ are injured or α is initialized.

Proof. When α initially defines $g_\alpha^{D_p}$ on those components in **Case 2** or **Case 3** of its strategy, it chooses $u_{\alpha,m}[s]$ large, and so $u_{\alpha,m}[s] > l_m[s]$.

Consider an α-stage $t > s$ and assume α has not been initialized and the components in $\mathcal{M}_i^{D_p}$ remain intact. Since the D_p-use on the edges in the components remains the same, we have $l_m[t] = l_m[s]$, and so $(D_p \restriction l_m[s])[t] = (D_p \restriction l_m[s])[s]$. In particular, any update of the value of n_α in **Case 2** or in **Step 1** of **Case 3** of the T_i^p-strategy does not cause n_α to fall below m. Therefore, $u_{\alpha,m}[t]$ either has the same value as in the previous α-stage, or is updated by the preliminary action of **Case 2** (and so is chosen large), or is redefined in **Step 2** of **Case 3** (to its value at the previous α-stage). In all cases, $u_{\alpha,m}[t] \geq u_{\alpha,m}[s]$. □

Lemma 1.3.12. Let α be an R_e^q-strategy that acts in **Case 2** at stage s by defining v_α. Let $t > s$ be the next α-stage and assume α is not initialized before t.

(1) At stage t, for all S- and T-strategies β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, f_β and $g_\beta^{D_p}[t]$ is defined on the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$. Furthermore, for T_i^p-strategies β, the minimum use of the computations $g_\beta^{D_p}[t]$ on the $(5n_\alpha+3)$- and $(5n_\alpha+4)$-loops is greater than v_α.

(2) If $q < p$, then the D_p-use for $g_\beta^{D_p}[t]$ on a_{2n_α}, $a_{2n_\alpha+1}$, the $(5n_\alpha+1)$-loops, $(5n_\alpha+2)$-loops, and the 2-loops in $\mathcal{G}$ is greater than v_α.

Proof. For (1), since α took the w_1 outcome at the last α-stage s and was not initialized after, it is now in **Case 3** at stage t. So in particular, it must be the case that α saw that for all S- and T-strategies β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$ that their parameters n_β have exceeded

n_α. In particular, if β is a T_i^p-strategy, it extended its map $g_\beta^{D_p}[s]$ on the $(5n_\alpha+3)$- and $(5n_\alpha+4)$-loops in $\mathcal{G}$ with a large use u_{β,n_α}. For each such β, we have that $u_{\beta,n_\alpha} > v_\alpha$. Furthermore, we claim that β cannot be challenged by another R-strategy before stage t. Suppose γ challenges β after β meets α's challenge. If $\beta^\frown\langle\infty\rangle \subseteq \gamma \subset \alpha$, then γ takes outcome w_1 when it challenges β, initializing α. If $\alpha \subseteq \gamma$, then γ cannot act until after stage t.

For (2), if β was a T_i^p-strategy where $q < p$ and $n_\beta > n_\alpha$, then α enumerated u_{β,n_α} into A_p, causing the map $g_\beta^{D_p}[s]$ to now be undefined on the entirety of the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$. At stage t when α is eligible to act again, we have that β must have recovered its map on a_{2n_α}, $a_{2n_\alpha+1}$, the $(5n_\alpha+1)$-loops, $(5n_\alpha+2)$-loops, and the 2-loops in $\mathcal{G}$ with a new large use $u_{\beta,n_\alpha} > v_\alpha$.

On the other hand, if $n_\beta \leq n_\alpha$, when α challenged β, then $g_\beta^{D_p}$ was not yet defined on any of the loops in the $2n\alpha$th and $(2n_\alpha+1)$st components. Therefore, when β defines $g_\beta^{D_p}$ on these components, it uses a large use on every loop. $\square$

Lemma 1.3.13. Let α be an R_e^q-strategy that acts in **Case 2** at stage s by defining v_α and challenging higher priority S and T-strategies. If t is the next α-stage and α has not been initialized, then $D_q[t] \upharpoonright \langle v_\alpha, q\rangle + 1 = D_q[s] \upharpoonright \langle v_\alpha, q\rangle + 1$. In particular, all of the loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{B}_q$ remain intact.

Proof. At stage s, α takes the w_1 outcome. By the proof of Lemma 1.3.8, no other strategy enumerates numbers at stage s. Since the strategies extending $\alpha^\frown\langle w_1\rangle$ and to the right of α choose new large parameters, none can enumerate a number below $\langle v_\alpha, q\rangle$ into D_q. If a strategy $\beta \subseteq \alpha$ enumerates a number, the path moves left, initializing α. Therefore, unless α is initialized, no number below $\langle v_\alpha, q\rangle$ can enter D_q before stage t. $\square$

Lemma 1.3.14. Let α be an R_e^q-strategy that acts in **Case 3** at stage s and takes the s outcome. Unless α is initialized, no number below the uses of the loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{B}_q$ or below m_q is enumerated into D_q after stage s.

Proof. The proof of this lemma is almost identical to the proof of Lemma 1.3.5. $\square$

We now state and prove the verification lemma for our construction.

Lemma 1.3.15 (Main Verification Lemma)**.** Let $TP = \liminf_s p_s$ be the true path of the construction, where p_s denotes the current true path at stage s of the construction. Let $\alpha \subset TP$.

(1) If α is an N_e^p-strategy, then there is an outcome o and an α-stage t_α such that for all α-stages $s \geq t_\alpha$, α takes outcome o where o ranges over $\{s, w_0\}$.

(2) If α is an S_e-strategy, then either α takes outcome ∞ infinitely often or there is an outcome w_n and a stage $\hat{t}$ such that for all α-stages $s > \hat{t}$, α takes outcome w_n. If $\mathcal{G} \cong \mathcal{M}_e$, then α takes the ∞ outcome infinitely often and α defines a partial embedding $f_\alpha : \mathcal{G} \to \mathcal{M}_e$ which can be extended to a computable isomorphism $\hat{f}_\alpha : \mathcal{G} \to \mathcal{M}_e$.

(3) Let α be a T_i^p-strategy. If $\mathcal{G} \cong \mathcal{M}_i^{D_p}$, then α takes the ∞ outcome infinitely often, and α defines a partial embedding $g_\alpha^{D_p} : \mathcal{G} \to \mathcal{M}_i^{D_p}$ which can be extended to a D_p-computable isomorphism $\hat{g}_\alpha^{D_p} : \mathcal{G} \to \mathcal{M}_i^{D_p}$.

(4) If α is an R_e^q-strategy, then there is an outcome o and an α-stage t_α such that for all α-stages $s \geq t_\alpha$, α takes outcome o where o ranges over $\{s, w_1, w_0\}$.

In addition, α satisfies its assigned requirement.

Proof. We first prove (1). Let $\alpha \subseteq TP$ be an N_e^p-strategy and let s_0 be the least stage such that for all $s \geq s_0$, $\alpha \leq_L p_s$. Let x_α be its parameter at stage s_0. Suppose that at every α-stage $s \geq s_0$, either $\Phi_e^{\overline{D_p}}(x_\alpha)[s] \uparrow$ or $\Phi_e^{\overline{D_p}}(x_\alpha)[s] \downarrow \neq 0$. Then, α takes the w_0 outcome at every α-stage $s \geq s_0$. Furthermore, $\Phi_e^{\overline{D_p}}(x_\alpha)$ either diverges or converges to a number other than 0. Since $x_\alpha \notin A_p$, α has met N_e^p.

Otherwise, there is some α-stage $t > s_0$ where $\Phi_e^{\overline{D_p}}(x_\alpha)[t] \downarrow = 0$. At stage t, α enumerates x_α into A_p and takes the s outcome. Since α is never initialized, it takes the s outcome at every α-stage $s \geq t$. Furthermore, by Lemma 1.3.5 we have that

$$\Phi_e^{\overline{D_p}}(x_\alpha) = \Phi_e^{\overline{D_p}}(x_\alpha)[t] = 0 \neq 1 = A_p(x_\alpha),$$

and so N_e^p is satisfied.

For (2), let $\alpha \subseteq TP$ be an S_e-strategy and let s_0 be the least stage such that for all $s \geq s_0$, $\alpha \leq_L p_s$. Suppose that α only takes the ∞ outcome finitely often. Fix an α-stage $s_1 > s_0$ such that α does not take the ∞ outcome at any α-stage $s \geq s_1$. Suppose that α takes the w_n outcome at stage s_1. There are two cases to consider.

If α is not challenged at stage s_1, then α acts as in **Case 3** of the S_e-strategy. Since α cannot be challenged by a requirement extending $\alpha^\frown\langle w_n\rangle$, α remains in **Case 3** at future α-stages unless it finds copies of the $2n$th and $(2n+1)$st components of $\mathcal{G}$ in $\mathcal{M}_e$. However, if it finds these copies, it would take the ∞ outcome, and so α must not ever find these copies. Hence, α takes the w_n outcome at every α-stage $s \geq s_1$. Moreover, $\mathcal{M}_e$ does not contain copies of the $2n$th and $(2n+1)$st components of $\mathcal{G}$ and so $\mathcal{M}_e$ is not isomorphic to $\mathcal{G}$.

If α is challenged at stage s_1, then α acts as in **Case 2** of the S_e-strategy. By a similar argument to the one above, α can never meet this challenge by finding images for the new loops in the $2n$th and $(2n+1)$st components of $\mathcal{G}$. Therefore, $\mathcal{M}_e \ncong \mathcal{G}$ and α takes the w_n outcome for all α-stages $s \geq s_1$.

From these two cases, it follows that if $\mathcal{G} \cong \mathcal{M}_e$, then α must take the ∞ outcome infinitely often. Let $n_\alpha[s]$ denote the value of n_α at the end of stage s (i.e., n_α could be possibly redefined by a challenging strategy during stage s). We claim that the value of $n_\alpha[s]$ goes to infinity as s goes to infinity. Suppose that α takes the ∞ outcome at a stage $s > s_0$. Either $n_\alpha[s] = n_\alpha[s-1]+1$ or $n_\alpha[s] = n_\beta[s]$ for some R-strategy β where $\alpha^\frown\langle\infty\rangle \subseteq \beta$ and $n_\beta[s] \leq n_\alpha[s-1]$. There can only be finitely many such β with $n_\beta[s] \leq n_\alpha[s-1]$. Once those strategies have challenged α (if ever), the value of n_α cannot drop below $n_\alpha[s-1]$ again. As α takes the ∞ outcome infinitely often, it follows that $n_\alpha[s]$ goes to infinity as s increases.

Let $f_\alpha = \bigcup\limits_{s \geq s_0} f_\alpha[s]$ be the limit of the partial $f_\alpha[s]$ embeddings for $s \geq s_0$. By Lemma 1.3.3, it remains to show that f_α can be computably extended to an embedding $\hat{f}_\alpha$ which is defined on all of $\mathcal{G}$. Note that no strategy $\beta \subseteq \alpha$ can add loops to $\mathcal{G}$ after stage s_0 or else the current true path would move to the left of α in the tree. Since $n_\alpha[s] \to \infty$ as $s \to \infty$, for each k, there is an α-stage s_k at which the n_α parameter starts with value k and α takes the ∞ outcome. At this stage, α found a copy of the $2k$th and $(2k+1)$st components of $\mathcal{G}$ (which consist of at least the initial set of loops) in $\mathcal{M}_e$ and defined $f_\alpha[s_k]$ on these loops. Therefore, f_α is defined on all of the initial loops attached to each a_{2k} and a_{2k+1}.

Only R-strategies β such that $\alpha^\frown\langle\infty\rangle \subseteq \beta$ can add loops to components on which f_α is already defined. If such a strategy β adds loops as in **Case 3** of its action, then it challenges α to find copies of these new loops. Since α takes the ∞ outcome infinitely often, it meets this challenge and extends f_α to be defined on the loops created by β. Then, β adds the homogenizing loops as in **Case 4** of its action. If $\mathcal{G} \cong \mathcal{M}_e$, then these homogenizing loops will have copies in $\mathcal{M}_e$. Suppose the $2n$th and $(2n+1)$st components of $\mathcal{G}$ have homogenizing loops, then because $\mathcal{G} \cong \mathcal{M}_e$, then only the nodes $f_\alpha(a_{2n})$ and $f_\alpha(a_{2n+1})$ in $\mathcal{M}_e$ will have copies of the respective homogenizing loops. So, we can computably extend f_α to $\hat{f}_\alpha$ by mapping the homogenizing loops to these copies, and thus $\hat{f}_\alpha$ is the computable isomorphism which satisfies the S_e requirement.

For (3), let $\alpha \subseteq TP$ be a T_i^p-strategy and let s_0 be the least stage such that $\alpha \leq_L p_s$ for all $s \geq s_0$. If $\mathcal{G} \ncong \mathcal{M}_i^{D_p}$, then we satisfy the T_i^p requirement trivially. So, suppose $\mathcal{G} \cong \mathcal{M}_i^{D_p}$.

We claim that α takes the ∞ outcome infinitely often and that $n_\alpha[s] \to \infty$. By Lemma 1.3.4, the true copies of each pair of components will eventually appear in $\mathcal{M}_i^{D_p}$ and become the oldest copies of these components. If $g_\alpha^{D_p}$ maps the $2m$th and $(2m+1)$st components of $\mathcal{G}$ to fake copies in $\mathcal{M}_i^{D_p}$, then by Lemma 1.3.11, when a number less than $l_m[s]$ enters D_p to remove the fake copies, then the map $g_\alpha^{D_p}[s]$ also becomes undefined on those components. Therefore, for each n, α will eventually define $g_\alpha^{D_p}[s]$ correctly on the $2n$th and $(2n+1)$st components of $\mathcal{G}$, mapping them to the true copies in $\mathcal{M}_i^{D_p}$.

For the same reason, α will also meet each challenge after s_0 by an R-strategy extending $\alpha^\frown\langle\infty\rangle$. It follows that $n_\alpha[s] \to \infty$ as $s \to \infty$ and that $g_\alpha^{D_p} = \bigcup\limits_{s \geq s_0} g_\alpha^{D_p}[s]$ will correctly map all loops in $\mathcal{G}$ into $\mathcal{M}_i^{D_p}$ except the homogenizing loops added in **Case 3** of an R-strategy.

It remains to show that we can extend $g_\alpha^{D_p}$ in a D_p-computable way to an embedding

$\hat{g}_\alpha^{D_p}$ defined on all of $\mathcal{G}$. Using the D_p oracle, we can tell when $g_\alpha^{D_p}[s]$ has correctly defined the original $(5m+1)$- and $(5m+2)$-loops from $\mathcal{G}$ into $\mathcal{M}_i^{D_p}$, as well as the $(5m+3)$- and $(5m+4)$-loops added (if ever) to $\mathcal{G}$ by an R-requirement. The D_p oracle will then tell us when the correct homogenizing loops show up in $\mathcal{M}_i^{D_p}$ (assuming that they were added to $\mathcal{G}$), so that we can extend $g_\alpha^{D_p}$ in a D_p-computable manner.

For (4), suppose $\alpha \subset TP$ is an R_e^q-strategy and let $n_\alpha = n$ be its parameter. Let s_0 be the least stage such that $\alpha \leq_L p_s$ for all $s \geq s_0$. If α remains in the first part of **Case 2** from its description for all α-stages $s \geq s_0$, then R_e^q is trivially satisfied because $\Phi_e^{D_q}$ is not an isomorphism between $\mathcal{G}$ and $\mathcal{B}_q$, and α takes the w_0 outcome cofinitely often.

Otherwise, there is an α-stage $s_1 > s_0$ such that $\Phi_e^{D_q}[s_1]$ maps the $2n$th and $(2n+1)$st components of $\mathcal{G}$ isomorphically into $\mathcal{B}_q$. Then, α carries out all actions described in the second part of **Case 2**. In particular, it defines its target number v_α *after* enumerating all uses $u_{\gamma,n}$ (if they exist) for any T_i^p-strategy γ of higher priority with $q < p$ in P. α's challenge will eventually be met by all β such that $\beta^\frown\langle\infty\rangle \subset \alpha$ since $\beta^\frown\langle\infty\rangle \subset TP$, and so let $s_2 > s_1$ be the next α-stage. By Lemma 1.3.13, $D_q[s_2] \upharpoonright \langle v_\alpha, q\rangle + 1 = D_q[s_1] \upharpoonright \langle v_\alpha, q\rangle + 1$, so the loops added to $\mathcal{B}_q$ at stage s_1 remain intact. At stage s_2, α enumerates v_α into A_q, moves the $(5n+3)$- and $(5n+4)$-loops in $\mathcal{B}_q$, and takes the s outcome at the end of this stage and at every future α-stage. By Lemma 1.3.12, for every T_i^p-strategy β with $\beta^\frown\langle\infty\rangle \subseteq \alpha$, we have that

- $g_\beta^{D_p}[s_2+1]$ is now undefined on the $(5n_\alpha+3)$- and $(5n_\alpha+4)$-loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$, and
- if $q < p$, then $g_\beta^{D_p}[s_2+1]$ is now undefined on the entirety of the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$.

Since $\alpha \subset TP$, α will never get initialized again. By Lemmas 1.3.13 and 1.3.14, we preserve $A_q \upharpoonright m_\alpha$. Recall that m_α is the use of the computation of $\Phi_e^{D_q}[s_1]$ where $\Phi_e^{D_q}[s_1](a_{2n}) = b_{2n}^q$ and $\Phi_e^{D_q}[s_1](a_{2n+1}) = b_{2n+1}^q$. So we have that $\Phi_e^{D_q} = \Phi_e^{D_q}[s_1]$, and so $\Phi_e^{D_q}(a_{2n}) = b_{2n}^q$ and

$\Phi_e^{D_q}(a_{2n+1}) = b^q_{2n+1}$. However, a_{2n} is connected to a cycle of length $5n+3$ whereas b_{2n} is connected to a cycle of length $5n+4$, and so $\Phi_e^{D_q}$ cannot be a D_q-computable isomorphism between $\mathcal{G}$ and $\mathcal{B}_q$. □

1.4 Embedding lattices

The techniques used in the proof of Theorem 1.1.10 to make $\mathcal{G}$ computably categorical or not relative to a degree are compatible with techniques to create minimal pairs of c.e. degrees. In fact, we can show that we can embed the four element diamond lattice into the c.e. degrees in the following way.

Theorem 1.4.1. There exists a computable computably categorical directed graph $\mathcal{G}$ and c.e. sets X_0 and X_1 such that

(1) X_0 and X_1 form a minimal pair,

(2) $\mathcal{G}$ is not computably categorical relative to X_0,

(3) $\mathcal{G}$ is computably categorical relative to X_1, and

(4) $\mathcal{G}$ is computably categorical relative to $X_0 \oplus X_1$.

Proof. We begin by listing the requirements needed for this construction.

$$S_i : \text{if } \mathcal{G} \cong \mathcal{M}_i, \text{ then there exists a computable isomorphism } f_i : \mathcal{G} \to \mathcal{M}_i$$

$$T_i : \text{if } \mathcal{G} \cong \mathcal{M}_i^{X_1}, \text{ then there exists an } X_1\text{-computable isomorphism } g_i^{X_1} : \mathcal{G} \to \mathcal{M}_i^{X_1}$$

$$R_e : \Phi_e^{X_0} : \mathcal{G} \to \mathcal{B} \text{ is not an isomorphism}$$

$$J_i : \text{if } \mathcal{G} \cong \mathcal{M}_i^{X_0 \oplus X_1}, \text{ then there exists an } (X_0 \oplus X_1)\text{-computable isomorphism}$$

$$h_i^{X_0 \oplus X_1} : \mathcal{G} \to \mathcal{M}_i^{X_0 \oplus X_1}$$

$$N_e : \text{if } \Phi_e^{X_0} = \Phi_e^{X_1} \text{ is total, then there exists a computable function } \Delta_e \text{ with } \Delta_e = \Phi^X.$$

Here, $\mathcal{B}$ is an X_0-computable graph that we will build alongside $\mathcal{G}$, similar to how we built the copies of $\mathcal{G}$ in the proof of Theorem 1.1.10. We also make a note that in the usual construction of a minimal pair, we would need requirements to ensure that X_0 and X_1 are not computable sets. However, assuming that we meet all the listed requirements above, these conditions are indirectly satisfied. In particular, X_0 cannot be computable since $\mathcal{G}$ is not computably categorical relative to X_0. X_1 also cannot be computable because if we assume that it is, then $X_0 \oplus X_1 \equiv_T X_0$, and so we have that $\mathcal{G}$ is computably categorical relative to X_0 if and only if it is computably categorical relative to $X_0 \oplus X_1$. But, by the R_e requirements, $\mathcal{G}$ is not computably categorical relative to X_0 and by the J_i requirements, it is computably categorical relative to $X_0 \oplus X_1$.

We will have largely the same strategies to meet the S_i requirements (section 1.2.2), the T_i and J_i requirements (section 1.2.3), and the R_e requirements (section 1.2.4). We will reintroduce these strategies with the new notation to match the requirements listed above. Afterwards, we detail the formal strategies for the new N_e requirements.

1.4.1 Strategies and outcomes

R_e-strategies and outcomes

We begin this section by outlining the formal strategies and outcomes for the R_e requirement. Compared to the strategy outlined in section 1.3.6, the R_e-strategies are simpler since they are only assigned a singular c.e. set X_0. That is, the X_0-uses become singular numbers and not numbers coded by a pair since there are no A_q or D_q sets.

We will also make similar adjustments to our $(X_0 \oplus X_1)$-uses by saying that if we have a use u for an $(X_0 \oplus X_1)$-computation that we want to enumerate into X_1 to destroy a computation, then we define the $(X_0 \oplus X_1)$-use to be $2u + 1$. Then by enumerating $2u + 1$ into $X_0 \oplus X_1$, we also enumerate u into X_1 to destroy the relevant computation.

Additionally, in **Case 2** below, when an R_e-strategy has to enumerate a use $u_{\gamma,n}$, the only such γ it needs to worry about are J_i-strategies γ where $\gamma^\frown\langle\infty\rangle \subseteq \alpha$. Here, the J_i-strategies play the role of the T_i^p-strategies for $q < p$ with $q = X_0$ and $p = X_0 \oplus X_1$ as in the previous construction.

Let α be an R_e-strategy eligible to act at stage s.

Case 1: If α is first eligible to act at stage s or has been initialized, define the parameter $n_\alpha = n$ to be large and take outcome w_0.

Case 2: If we are not in **Case 1** and α took outcome w_0 at the last α-stage, check whether $\Phi_e^{X_0}[s]$ maps the $2n$th and $(2n+1)$st components of $\mathcal{G}$ isomorphically into $\mathcal{B}$. If not, take outcome w_0.

If so, set m_α to be the maximum X_0-use of these computations. Let v_α be large. Add a $(5n+3)$-loop to a_{2n} in $\mathcal{G}$ and to b_{2n} in $\mathcal{B}$ (with X_0-use v_α) and add a $(5n+4)$-loop to a_{2n+1} in $\mathcal{G}$ and to b_{2n+1} in $\mathcal{B}$ (with X_0-use v_α).

For each J-strategy γ where $\gamma^\frown\langle\infty\rangle \subseteq \alpha$, enumerate the use $u_{\gamma,n}$ into X_1 (and hence $2u_{\gamma,n} + 1$ into $X_0 \oplus X_1$), and challenge γ. Note that if $n_\gamma < n_\alpha$, then there is no $u_{\gamma,n}$ to enumerate into X_1. For each T- and S-strategy β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, challenge β and reset $n_\beta = n_\alpha$ if $n_\beta > n_\alpha$. Otherwise, leave n_β as it is. For each J-strategy β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, reset n_β to be the least $m \leq n_\alpha$ such that $h_\beta^{X_0 \oplus X_1}$ does not match all of the $2m$th and $(2m+1)$st components of $\mathcal{G}$. Take outcome w_1.

Case 3: If α took outcome w_1 at the last α-stage, then enumerate v_α into X_0, move the $(5n+3)$-loop in $\mathcal{B}$ from b_{2n} to b_{2n+1}, and move the $(5n+4)$-loop in from b_{2n+1} to b_{2n}. Attach a $(5n+1)$-loop to a_{2n+1} and b_{2n+1} and a $(5n+2)$-loop to a_{2n} and b_{2n}. Let the X_0-use of these new loops in $\mathcal{B}$ be large. Take outcome s.

Case 4: If α took outcome s at the last α-stage and has not been initialized, then take outcome s.

T_i-strategies and outcomes

In the previous construction for Theorem 1.1.10, we had the T_i^p-strategies where p is an element of the poset $P = (P, \leq)$. For the minimal pair construction, we have that the T_i and

J_i-strategies correspond to T_i^p-strategies where $p = X_1$ in the former and $p = X_0 \oplus X_1$ in the latter. We now state the formal strategies for the T_i requirements, with comments on any differences from the previous construction.

Let α be a T_i-strategy eligible to act at stage s. Recall that the T_i requirement is to ensure that $\mathcal{G}$ is computably categorical relative to our c.e. set X_1.

Case 1: If α is acting for the first time or has been initialized since the last α-stage, set $n_\alpha = 0$, define $g_\alpha^{X_1}[s]$ to be the empty function, and take the w_0 outcome.

Case 2: α is currently challenged by an R_e-strategy β where $\alpha^\frown\langle\infty\rangle \subseteq \beta$. Let s_0 be the stage at which β challenged α. When β challenged α at stage s_0, it redefined $n_\alpha = n_\beta$.

We now perform the main action in this case. If $g_\alpha^{X_1}$ is already defined on the $(5n_\alpha+1)$- and $(5n_\alpha+2)$-loops of the $2n_\alpha$th and $(2n_\alpha+1)$st components in $\mathcal{G}$, then α searches for the oldest and lexicographically least copies of the $(5n_\alpha+3)$- and $(5n_\alpha+4)$-loops in $\mathcal{M}_i^{X_1}[s]$. If $g_\alpha^{X_1}$ is not currently defined on any of the loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$, then α searches for the oldest and lexicographically least copies of these components in $\mathcal{M}_i^{X_1}[s]$. In either case, if such copies are found, extend $g_\alpha^{X_1}[s]$ to map onto these copies with a large X_1-use u_{α,n_α}, increment n_α by 1, and check if $n_\alpha > n_\beta$ for this new n_α. If yes, take the ∞ outcome and declare β's challenge to α to be met. If not, then take the w_{n_α} outcome and let β's challenge to α remain active.

Case 3: α is not currently challenged by an R_e-strategy. Let t be the last α-stage. In this case, α defined $g_\alpha^{X_1}[t]$ on the $2m$th and $(2m+1)$st components with X_1-uses $u_{\alpha,m}$ for $m < n_\alpha$. Let l_m be the max X_1-use for the computation of a loop in the image of the $2m$th and $(2m+1)$st components under $g_\alpha^{X_1}[t]$.

Step 1: If there is an $m < n_\alpha$ such that $X_1[t] \upharpoonright l_m \neq X_1[s] \upharpoonright l_m$, then let m be the least such value. Note that for $m \leq m^* < n_\alpha$, the map $g_\alpha^{X_1}$ is now undefined on the $2m^*$th and $(2m^*+1)$st components of $\mathcal{G}$. The loops in the image of the $2k$th and $(2k+1)$st components of $\mathcal{G}$ under $g_\alpha^{X_1}[t]$ for $k < m$ remain in $\mathcal{M}_i^{X_1}[s]$. Update $n_\alpha = m$.

Step 2: By the update in **Step 1**, we have that for each $m < n_\alpha$ that $X_1[t] \upharpoonright l_m = X_1[s] \upharpoonright$

l_m. For each $m < n_\alpha$, if any, where $X_1[t] \upharpoonright u_{\alpha,m} \neq X_1[s] \upharpoonright u_{\alpha,m}$, set $g_\alpha^{X_1}[s] = g_\alpha^{X_1}[t]$ on the loops in $\mathcal{G}$ in the $2m$th and $(2m+1)$st components with the same X_1-use as at stage t.

Step 3: We can now perform the main action of this case. α searches for the oldest and lexicographically least copies of the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$ in $\mathcal{M}_i^{X_1}[s]$. If no copies are found, leave $g_\alpha^{X_1}$ and n_α unchanged and take outcome w_{n_α}. Otherwise, extend $g_\alpha^{X_1}$ by mapping the loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$ to their copies in $\mathcal{M}_i^{X_1}$ with an X_1-use u_{α,n_α} where u_{α,n_α} is chosen large, increment n_α by 1, and take the ∞ outcome.

We make a note that for the X_1-uses u defined via the T_i-strategy have a corresponding $(X_0 \oplus X_1)$-use which is $2u+1$. So, in the case where the use u must be enumerated into X_1, we can do this by enumerating $2u+1$ into $X_0 \oplus X_1$.

Note that the change in X_1 in **Case 3** can be caused by an R-strategy issuing a challenge to some higher priority J-strategy. Unlike in the poset case, however, any X_0-changes will not cause changes in any $\mathcal{M}_i^{X_1}$ since X_0 and X_1 are incomparable c.e. sets, and so we do not need our T_i-strategies to lift its uses to account for any diagonalizations performed by some R-strategy. That is, the preliminary step in **Case 2** in section 1.3.5 is removed for the T_i-strategies in the minimal pair construction.

J_i-strategies and outcomes

We now give the formal description of the J_i-strategies and outcomes. In particular, these strategies must account for any diagonalizations in $\mathcal{B}$ caused by some lower priority R-strategy since $\mathcal{B}$ is an X_0- and thus $(X_0 \oplus X_1)$-computable directed graph.

Let α be a J_i-strategy eligible to act at stage s. Recall that the J_i requirement is to ensure that $\mathcal{G}$ is computably categorical relative to $X_0 \oplus X_1$.

Case 1: If α is acting for the first time or has been initialized since the last α-stage, set $n_\alpha = 0$, define $h_\alpha^{X_0 \oplus X_1}[s]$ to be the empty function, and take the w_0 outcome.

Case 2: α is currently challenged by an R_e-strategy β where $\alpha^\frown\langle\infty\rangle \subseteq \beta$. Let s_0 be the stage at which β challenged α. When β challenged α at stage s_0, it redefined n_α to equal the least $m \leq n_\beta$ such that $h_\alpha^{X_0 \oplus X_1}$ is not fully defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$.

If s is the first α-stage since s_0 and n_α was greater than n_β at stage s_0, then we have to perform a preliminary action. In this case, β enumerated u_{α,n_β} into X_1 (and so $2u_{\alpha,n_\beta}+1$ is enumerated into $X_0 \oplus X_1$), causing the map $h_\alpha^{X_0 \oplus X_1}[s_0]$ to become undefined on the $2n_\beta$th and $(2n_\beta+1)$st components of $\mathcal{G}$. Redefine $h_\alpha^{X_0 \oplus X_1}[s]$ to be equal to $h_\alpha^{X_0 \oplus X_1}[s_0]$ on the 2-loops, $(5n_\beta+1)$-loops, and $(5n_\beta+2)$-loops in these components, and choose a new large number $2u_{\alpha,n_\beta}+1$ as the $(X_0 \oplus X_1)$-use for this computation. This ends the preliminary step.

We now perform the main action in this case. If $n_\alpha = n_\beta$ and $h_\alpha^{X_0 \oplus X_1}$ is already defined on the $(5n_\alpha+1)$- and $(5n_\alpha+2)$-loops of the $2n_\alpha$th and $(2n_\alpha+1)$st components in $\mathcal{G}$, then α searches for the oldest and lexicographically least copies of the $(5n_\alpha+3)$- and $(5n_\alpha+4)$-loops in $\mathcal{M}_i^{X_0 \oplus X_1}[s]$. If $h_\alpha^{X_0 \oplus X_1}$ is not currently defined on any of the loops in the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{G}$, then α searches for the oldest and lexicographically least copies of these components in $\mathcal{M}_i^{X_0 \oplus X_1}[s]$. In either case, if such copies are found, extend $h_\alpha^{X_0 \oplus X_1}[s]$ to map onto these copies with a large use $2u_{\alpha,n_\alpha}+1$, increment n_α by 1, and check if $n_\alpha > n_\beta$ for this new n_α. If yes, take the ∞ outcome and declare β's challenge to α to be met. If not, then take the w_{n_α} outcome and let β's challenge to α remain active.

Case 3: α is not currently challenged by an R_e-strategy. Let t be the last α-stage. In this case, α defined $h_\alpha^{X_0 \oplus X_1}[t]$ on the $2m$th and $(2m+1)$st components with use $u_{\alpha,m}$ for $m < n_\alpha$. Let l_m be the max $(X_0 \oplus X_1)$-use for the computation of a loop in the image of the $2m$th and $(2m+1)$st components under $h_\alpha^{X_0 \oplus X_1}[t]$.

Step 1: If there is an $m < n_\alpha$ such that $(X_0 \oplus X_1)[t] \upharpoonright l_m \neq (X_0 \oplus X_1)[s] \upharpoonright l_m$, then let m be the least such value. Note that for $m \leq m^* < n_\alpha$, the map $h_\alpha^{X_0 \oplus X_1}$ is now undefined on the $2m^*$th and $(2m^*+1)$st components of $\mathcal{G}$. The loops in the image of the $2k$th and $(2k+1)$st components of $\mathcal{G}$ under $h_\alpha^{X_0 \oplus X_1}[t]$ for $k < m$ remain in $\mathcal{M}_i^{X_0 \oplus X_1}[s]$. Update $n_\alpha = m$.

Step 2: By the update in **Step 1**, we have that for each $m < n_\alpha$ that $(X_0 \oplus X_1)[t] \upharpoonright l_m =$

$(X_0 \oplus X_1)[s] \upharpoonright l_m$. For each $m < n_\alpha$, if any, where $(X_0 \oplus X_1)[t] \upharpoonright u_{\alpha,m} \neq (X_0 \oplus X_1)[s] \upharpoonright u_{\alpha,m}$, set $h_\alpha^{X_0 \oplus X_1}[s] = h_\alpha^{X_0 \oplus X_1}[t]$ on the loops in $\mathcal{G}$ in the $2m$th and $(2m+1)$st components with the same use as at stage t.

Step 3: We can now perform the main action of this case. α searches for the oldest and lexicographically least copies of the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{G}$ in $\mathcal{M}_i^{X_0 \oplus X_1}[s]$. If no copies are found, leave $h_\alpha^{X_0 \oplus X_1}$ and n_α unchanged and take outcome w_{n_α}. Otherwise, extend $h_\alpha^{X_0 \oplus X_1}$ by mapping the loops in the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{G}$ to their copies in $\mathcal{M}_i^{X_0 \oplus X_1}$ with use $2u_{\alpha,n_\alpha} + 1$ where u_{α,n_α} is chosen large, increment n_α by 1, and take the ∞ outcome.

Unlike the T_i-strategies, we need to perform the preliminary step in **Case 2** for the J_i-strategies to account for any diagonalizations performed by an R-strategy on $\mathcal{B}$-components. This preliminary steps allow our J_i-strategies α to lift their uses so that even if the positions of the $(5n_\beta + 3)$- and $(5n_\beta + 4)$-loops in $\mathcal{B}$ are switched by some R-strategy β, α is able to redefine $h_\alpha^{X_0 \oplus X_1}$ on the affected $\mathcal{G}$-components.

N_e-strategies and outcomes

The N_e requirements are required so that X_0 and X_1 form a minimal pair, and we will use the standard strategy to satisfy each N_e. Let α be an N_e-strategy eligible to act at stage s.

Case 1: If α is first eligible to act at stage s or was initialized since the last α-stage, define its parameter $n_\alpha = 0$ and take the w_0 outcome.

Case 2: If α took the w_{n_α} outcome at the previous α-stage, check if

$$\Phi_e^{X_0} \upharpoonright (n_\alpha + 1) = \Phi_e^{X_1} \upharpoonright (n_\alpha + 1).$$

If the equality does not hold, continue taking the w_{n_α} outcome with the n_α parameter unchanged. If the equality holds, set $\Delta_e(n_\alpha) = \Phi_e^{X_0}(n_\alpha)$, increment n_α by 1, and take the ∞ outcome.

In the verification, we will show that when we define $\Delta_e(n_\alpha) = \Phi_e^{X_0}(n_\alpha)$ at a stage s, then for any stage $t > s$, at least one of $\Phi_e^{X_0}(n_\alpha)$ or $\Phi_e^{X_1}(n_\alpha)$ is defined and equal to $\Delta_e(n_\alpha)$.

1.4.2 Construction

Let $\Lambda = \{\infty <_\Lambda \cdots <_\Lambda w_2 <_\Lambda s <_\Lambda w_1 <_\Lambda w_0\}$ be the set of outcomes, and let $T = \Lambda^{<\omega}$ be our tree of strategies. The construction will be performed in ω many stages s.

We define the **current true path** p_s, the longest strategy eligible to act at stage s, inductively. For every s, λ, the empty string, is eligible to act at stage s. Suppose the strategy α is eligible to act at stage s. If $|\alpha| < s$, then follow the action of α to choose a successor $\alpha^\frown\langle o\rangle$ on the current true path. If $|\alpha| = s$, then set $p_s = \alpha$. For all strategies β such that $p_s <_L \beta$, initialize β (i.e., set all parameters associated to β to be undefined). If $\beta <_L p_s$ and $|\beta| < s$, then β retains the same values for its parameters.

1.4.3 Verification

Before we prove the main verification lemma for this construction, we first prove the following important lemma regarding the N_e-strategies.

Lemma 1.4.2. At most one strategy α enumerates numbers at any stage.

Proof. Suppose α is the highest priority strategy at stage s which enumerates a number into a set. Then α is an R-strategy that either takes the w_1 or s outcome for the first time. In both cases, all strategies extending α on either the w_1 or s outcome will act by defining their parameters to be large and taking the w_0 outcome. Hence, α is the only strategy which enumerates a number at stage s. □

Lemma 1.4.2 allows the N_e-strategies to succeed because at every stage, we can only at most lose one of the computations $\Phi_e^{X_0}(n)$ or $\Phi_e^{X_1}(n)$ for each n. So if for an N_e-strategy α we have that if $\Phi_e^{X_0} = \Phi_e^{X_1}$ is total, then the Δ_e will be defined for all $n \in \omega$ at the end of the construction.

Before we state and prove the main verification lemma, we would like to comment on how many of the key lemmas proven in section 1.3.7 carry over to this construction with as analogous lemmas with largely the same proofs. For example, we have a version of Lemma 1.3.5 for the P_e-strategies which holds by virtually the same proof. We also have analogues of Lemmas 1.3.9, 1.3.10, and 1.3.11 for our T_i- and J_i-strategies in this construction.

We now state and prove the main verification lemma for this construction.

Lemma 1.4.3 (Main Verification Lemma)**.** Let $TP = \liminf_s p_s$ be the true path of the construction, where p_s denotes the current true path at stage s of the construction. Let $\alpha \subset TP$.

(1) If α is an S_i-strategy, then either α takes outcome ∞ infinitely often or there is an outcome w_n and a stage t_α such that for all α-stages $s > t_\alpha$, α takes outcome w_n. If $\mathcal{G} \cong \mathcal{M}_i$, then α takes the ∞ outcome infinitely often and α defines a partial embedding $f_\alpha : \mathcal{G} \to \mathcal{M}_i$ which can be extended to a computable isomorphism $\hat{f}_\alpha : \mathcal{G} \to \mathcal{M}_i$.

(2) If α is a T_i-strategy or J_i-strategy, then either α takes outcome ∞ infinitely often or there is an outcome w_n and a stage t_α such that for all α-stages $s > t_\alpha$, α takes outcome w_n. If $\mathcal{G} \cong \mathcal{M}_i^{X_1}$ ($\mathcal{G} \cong \mathcal{M}_i^{X_0 \oplus X_1}$), then α takes the ∞ outcome infinitely often, and defines a partial embedding $g_\alpha^{X_1} : \mathcal{G} \to \mathcal{M}_i^{X_1}$ ($h_\alpha^{X_0 \oplus X_1} : \mathcal{G} \to \mathcal{M}_i^{X_0 \oplus X_1}$) which can be extended to an X_1-computable isomorphism $\hat{g}_\alpha^{X_1} : \mathcal{G} \to \mathcal{M}_i^{X_1}$ (($X_0 \oplus X_1$)-computable isomorphism $\hat{h}_\alpha^{X_0 \oplus X_1} : \mathcal{G} \to \mathcal{M}_i^{X_0 \oplus X_1}$).

(3) If α is an R_e-strategy, then there is an outcome o and an α-stage t_α such that for all α-stages $s \geq t_\alpha$, α takes outcome o where o ranges over $\{s, w_1, w_0\}$.

(4) If α is an N_e-strategy, then either α takes the ∞ outcome infinitely often or there is an outcome w_n and an α-stage t_α such that for all α-stages $s > t_\alpha$, α takes outcome w_n. In the former, α will define a total computable function Δ_e such that $\Delta_e(n) = \Phi^{X_0}(n)$ for all $n \in \omega$.

In addition, α satisfies its assigned requirement.

Proof. The arguments for (1)-(3) are similar to the arguments given for the proof of Lemma 1.3.15(2)-(4), so we only give the proof for (1) and (4).

For (4), let $\alpha \subseteq TP$ be an N_e-strategy, and let s_0 be the least stage such that for all $s \geq s_0$, $\alpha \leq_L p_s$. Since α is not initialized after stage s_0, the value of its parameter n_α can only increase. So if α takes the w_n outcome at a stage $t > s_0$, it will never take a w_l outcome for $l < n$ after stage t. Since α must take the ∞ outcome between taking distinct w_n outcomes, it follows that either α takes the ∞ outcome infinitely often or there is an n such that α takes the w_n outcome at cofinitely many α-stages.

If we have that $\Phi_e^{X_0}(k) \neq \Phi_e^{X_1}(k)$ for some k, then there exists some α-stage $s' \geq s_0$ such that for all $t \geq s'$,

$$\Phi_e^{X_0}[t] \upharpoonright (k+1) \neq \Phi_e^{X_1}[t] \upharpoonright (k+1).$$

After stage s', α can never take the ∞ outcome after taking the w_n outcome for any $n \geq k$. Therefore, α can only take the ∞ outcome finitely often and must take some w_n outcome at cofinitely many α-stages. N_e is trivially satisfied in this case.

If $\Phi_e^{X_0} = \Phi_e^{X_1}$ is total, then for each n, there is a stage t_n where

$$\Phi_e^{X_0}[t_n] \upharpoonright (n+1) = \Phi_e^{X_1}[t_n] \upharpoonright (n+1),$$

and also we have that

$$X_0[t_n] \upharpoonright (\text{use}(\Phi_e^{X_0}(n)) + 1) = X_0 \upharpoonright (\text{use}(\Phi_e^{X_0}(n)) + 1)$$

and

$$X_1[t_n] \upharpoonright (\text{use}(\Phi_e^{X_1}(n)) + 1) = X_1 \upharpoonright (\text{use}(\Phi_e^{X_1}(n)) + 1).$$

In this case, α takes the ∞ outcome infinitely often, and so it defined $\Delta_e(n)$ for all n. We now must verify that $\Delta_e(n) = \Phi_e^{X_0}(n)$ for all n. We claim that once $\Delta_e(n)$ had been defined at some stage t, then it will equal at least one of $\Phi_e^{X_0}(n)$ or $\Phi_e^{X_1}(n)$ at any stage after t.

Let $s_n \geq s_0$ be the first stage at which α defined its parameter to be equal to n. At any stage $s \geq s_n$, only R-strategies $\beta \supseteq \alpha^\frown\langle\infty\rangle$ have the potential to destroy a computation $\Phi_e^{X_i}(m)$ for $m < n$ that exists at stage s_n. This is because all $\beta <_L \alpha$ can no longer enumerate numbers into either X_i since $\alpha \leq_L p_s$ for all $s \geq s_0$ (and hence for all $s \geq s_n$), and all β such that $\alpha <_L \beta$ will pick their parameters and witnesses larger than the restraint placed on either X_i. By Lemma 1.4.2, we have that at any stage, at most one number is enumerated into either X_0 or X_1 (but not both). Hence, for all $m < n$, at least one of the computations $\Phi_e^{A_i}(m)$ will be preserved from stage s_n to s_{n+1}. Since $\Phi_e^{A_0} = \Phi_e^{A_1}$ is total, then $\Delta_e(n) = \Phi_e^{A_0}(n)$ for all n by the above argument. Thus, N_e is satisfied, and this finishes our proof of Theorem 1.4.1. □

□

1.5 Open questions

We end this chapter with some questions. One such question is whether we can embed bigger lattices into the c.e. degrees where elements of the lattice are partitioned like in the poset case. We restrict to distributive lattices since those are embeddable into the c.e. degrees.

Question 1.5.1. Let $L = (L, \wedge, \vee)$ be a computable distributive lattice and suppose we have a computable partition $L = L_0 \sqcup L_1$. Does there exist a computable computably categorical structure $\mathcal{A}$ and an embedding h of L into the c.e. degrees where $\mathcal{A}$ is computably categorical relative to each degree in $h(L_0)$ and is not computably categorical relative to each degree in $h(L_1)$?

Another direction is to look for restrictive cases where our techniques from this chapter do not work, which may entail looking outside of the c.e. degrees. A concrete question in this direction is whether the result of Downey, Harrison-Trainor, and Melnikov about $\mathbf{0}''$ can be improved.

Question 1.5.2. Is there a degree $\mathbf{d} < \mathbf{0}''$ such that if a computable structure $\mathcal{A}$ is computably categorical relative to $\mathbf{d}$, then for all $\mathbf{c} > \mathbf{d}$, $\mathcal{A}$ is computably categorical relative to $\mathbf{c}$?

Chapter 2

Extensions of categoricity relative to a degree

2.1 Categoricity relative to a generic degree

Most of this chapter is devoted to a result concerning computable categoricity relative to a generic degree. To motivate this result, we need some preliminary definitions.

Definition 2.1.1. A computable structure $\mathcal{A}$ is **d-computably categorical** if for all computable $\mathcal{B} \cong \mathcal{A}$, there exists a $\mathbf{d}$-computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Definition 2.1.2. A structure $\mathcal{A}$ has **degree of categoricity d** if $\mathcal{A}$ is $\mathbf{d}$-computably categorical and for all $\mathbf{c}$, if $\mathcal{A}$ is $\mathbf{c}$-computably categorical, then $\mathbf{d} \leq \mathbf{c}$. A degree $\mathbf{d}$ is a **degree of categoricity** if there is some structure with degree of categoricity $\mathbf{d}$.

Finding a characterization of degrees of categoricity in the Turing degrees has recently been an active topic in computable structure theory. For a survey paper of development up until 2017, see [12]. Degrees which are *not* degrees of categoricity exist, with Anderson and Csima producing several examples in [1]. One important example is the following.

Theorem 2.1.3 (Anderson, Csima [1])**.** There is a Σ_2^0 degree that is not a degree of categoricity.

In fact, the Σ_2^0 degree that they built to witness this result turns out to be low for isomorphism.

Definition 2.1.4. A degree $\mathbf{d}$ is **low for isomorphism** if for every pair of computable structures $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}$ is $\mathbf{d}$-computably isomorphic to $\mathcal{B}$ if and only if $\mathcal{A}$ is computably isomorphic to $\mathcal{B}$.

There are currently no known characterizations for LFI degrees, but they are connected to the generic degrees. We quickly recall the definition of an n-generic set.

Definition 2.1.5. A set A is **n-generic** if for all Σ_n^0 set of strings $S \subseteq 2^{<\omega}$, there exists an m such that either $A \upharpoonright m \in S$ or for all $\tau \supseteq A \upharpoonright m$, $\tau \notin S$. A degree $\mathbf{d}$ is **n-generic** if it contains an n-generic set.

Theorem 2.1.6 (Franklin, Solomon [13])**.** Every 2-generic degree is low for isomorphism.

This tells us that for a 2-generic degree $\mathbf{d}$, there exists *no* computable structure $\mathcal{A}$ where $\mathcal{A}$ is not computably categorical but is computably categorical relative to $\mathbf{d}$, one of the first examples of a degree where techniques from [30] do not apply. Here, we will show that this is optimal in the generic degrees, i.e., we can build a 1-generic degree $\mathbf{d}$ and a computable structure $\mathcal{A}$ which is not computably categorical but is computably categorical relative to $\mathbf{d}$.

Theorem 2.1.7. There exists a (properly) 1-generic G such that there is a computable directed graph $\mathcal{A}$ where $\mathcal{A}$ is not computably categorical but is computably categorical relative to G.

2.2 Informal strategies for Theorem 2.1.7

We first establish the informal strategies to meet three goals for our construction: to build a 1-generic set G, to make our graph $\mathcal{A}$ *not* computably categorical, and to make our graph $\mathcal{A}$ computably categorical relative to G. We will then describe the interactions that occur when we use all three strategies together, and how to resolve any issues that arise.

2.2.1 Building a 1-generic G

We will build a 1-generic G via infinitely many strategies. Recall that for a set G to be 1-generic, we must have that for all $e \in \omega$, there exists an initial segment σ of G where either $\sigma \in W_e$ or for all extensions $\tau \supseteq \sigma$, $\tau \notin W_e$. That is, either G meets or avoids each c.e. set.

For each $j \in \omega$, we meet the requirement

$$R_j : (\exists \sigma \subseteq G)(\sigma \in W_j \vee (\forall \tau \supseteq \sigma)(\tau \notin W_j)).$$

We will define sets $G[s]$, where s is a stage number, which are a Δ^0_2 approximation to our 1-generic G. Let α be the highest priority R-strategy. When it is first eligible to act at a stage s_0, α will define a parameter $n_\alpha > 0$ large (and so in particular, $n_\alpha > s_0$), and its goal is to find an extension $\tau_0 \supseteq G[s_0] \upharpoonright n_\alpha$ where $\tau_0 \in W_j$. At each α-stage, it searches for such an extension.

If α never finds such an extension, then we have that all extensions of $G[s_0] \upharpoonright n_\alpha$ avoid W_j and α succeeds trivially. If α finds an extension τ_0 such that $\tau_0 \in W_j$ at a stage $s_1 \geq s_0$, then α will define $G[s_1] = \tau_0 ^\frown 0^\omega$ and will initialize all lower priority strategies. In particular, all lower priority R-strategies will now have to redefine new, larger parameters (and so these parameters will be bigger than $|\tau_0|$) and must now use $G[s_1]$ as the current approximation of G at the end of stage s_1 for all stages $s \geq s_1$. We also have that $G[s_1] \upharpoonright n_\alpha = G[s_0] \upharpoonright n_\alpha$, so α does not cause changes on numbers below n_α.

If all R_j-strategies succeeded, then we define $G = \lim_s G[s]$ to be our 1-generic set. This limit exists by the observation in the previous paragraph.

2.2.2 Making $\mathcal{A}$ not computably categorical

We will build our directed graph $\mathcal{A}$ in stages. At stage $s = 0$, we set $\mathcal{A} = \emptyset$. Then, at stage $s > 0$, we add two new connected components to $\mathcal{A}[s]$ by adding the root nodes a_{2s} and a_{2s+1} for those components, and attaching to each node a 2-loop (a cycle of length 2). We then attach a $(5s + 1)$-loop to a_{2s} and a $(5s + 2)$-loop to a_{2s+1}. This gives us the configuration of loops:

$$a_{2s} : 2, 5s + 1$$
$$a_{2s+1} : 2, 5s + 2.$$

The connected component consisting of the root node a_{2s} with its attached loops will be referred to as the **2sth connected component** of $\mathcal{A}$. During the construction, we might add more loops to connected components of $\mathcal{A}$, which causes them to have one of the two following configurations:

$$a_{2s} : 2, 5s + 1, 5s + 2, 5s + 3$$
$$a_{2s+1} : 2, 5s + 1, 5s + 2, 5s + 4$$

or

$$a_{2s} : 2, 5s + 1, 5s + 2, 5s + 3, 5s + 4$$
$$a_{2s+1} : 2, 5s + 1, 5s + 2, 5s + 3, 5s + 4.$$

Note that the last configuration has that the $2s$th and $(2s + 1)$st components of $\mathcal{A}$ are isomorphic, which may be necessary as a result of a special interaction between strategies of all three types of requirements in this construction (see 2.2.4).

In order to make $\mathcal{A}$ not computably categorical, it is sufficient to construct a computable copy $\mathcal{B}$ such that for all $e \in \omega$, the computable function Φ_e is not an isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Similarly to $\mathcal{A}$, we build the directed graph $\mathcal{B}$ in stages. At stage $s = 0$, we set $\mathcal{B} = \emptyset$. At stage $s > 0$, we add root nodes b_{2s} and b_{2s+1} to $\mathcal{B}$ and attach to each one a 2-loop. Next, we attach a $(5s + 1)$-loop to b_{2s} and a $(5s + 2)$-loop to b_{2s+1}. However, throughout the construction, we may add new loops to specific components of $\mathcal{B}$. For the $2s$th and $(2s + 1)$st components of $\mathcal{B}$, we have three possible final configurations of the loops. If we never start the process of diagonalizing using these components, then they will remain the same forever:

$$b_{2s} : 2, 5s + 1$$

$$b_{2s+1} : 2, 5s + 2.$$

If we use these components to diagonalize against a computable map Φ_e, they will end in the following configuration:

$$b_{2s} : 2, 5s + 1, 5s + 2, 5s + 4$$

$$b_{2s+1} : 2, 5s + 1, 5s + 2, 5s + 3.$$

If these components were used to diagonalize against Φ_e and then certain interactions occur between all three types of requirements, these components may have the final configuration:

$$b_{2s} : 2, 5s + 1, 5s + 2, 5s + 3, 5s + 4$$

$$b_{2s+1} : 2, 5s + 1, 5s + 2, 5s + 3, 5s + 4.$$

For all $e \in \omega$, we meet the following requirement

$$P_e : \Phi_e : \mathcal{A} \to \mathcal{B} \text{ is not an isomorphism.}$$

To satisfy this requirement, we will diagonalize against Φ_e. Let α be a P_e-strategy.

When α is first eligible to act, it picks a large number n_α, and for the rest of this strategy, let $n = n_\alpha$. This parameter indicates which connected components of $\mathcal{B}$ will be used in the diagonalization. At future stages, α checks if Φ_e maps the $2n$th and $(2n+1)$st connected component of $\mathcal{A}$ to the $2n$th and $(2n+1)$st connected component of $\mathcal{B}$, respectively. If not, α does not take any action. If α sees such a computation, it acts in the following way.

At this point, our connected components in $\mathcal{A}[s]$ and $\mathcal{B}[s]$ are as follows:

$$a_{2n} : 2, 5n + 1 \quad b_{2n} : 2, 5n + 1$$

$$a_{2n+1} : 2, 5n + 2 \quad b_{2n+1} : 2, 5n + 2.$$

Since Φ_e looks like a potential isomorphism between $\mathcal{A}$ and $\mathcal{B}$, α will now take action to eventually force the true isomorphism to match a_{2n} with b_{2n+1} and to match a_{2n+1} with b_{2n}.

α adds a $(5n+2)$- and $(5n+3)$-loop to a_{2n} and a $(5n+1)$- and $(5n+4)$-loop to a_{2n+1} in $\mathcal{A}[s]$. It also attaches a $(5n+2)$- and $(5n+4)$-loop to b_{2n} and a $(5n+1)$- and $(5n+3)$-loop to b_{2n+1} in $\mathcal{B}$[s]. Our connected components in $\mathcal{A}[s]$ and in $\mathcal{B}[s]$ are now:

$$a_{2n} : 2, 5n+1, 5n+2, 5n+3 \qquad b_{2n} : 2, 5n+1, 5n+2, 5n+4$$

$$a_{2n+1} : 2, 5n+1, 5n+2, 5n+4 \qquad b_{2n+1} : 2, 5n+1, 5n+2, 5n+3.$$

For all higher priority S-strategies β such that $\beta^\frown\langle\infty\rangle \subseteq \alpha$, α enumerates the use u_{β,n_α} into G and sets $n_\beta = n_\alpha$. This enumeration occurs because if we have that $\beta^\frown\langle\infty\rangle \subseteq \alpha$, then α believes β will define a total function f_β^G on $\mathcal{A}$. Therefore, α doesn't restart when β extends its definition of f_β^G and so this map may be defined on the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ when α acts. In this case, $f_\beta^G[s-1]$ maps the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ to their copies in $\mathcal{M}_i^G$ with some use u_{β,n_α}. We will choose this use carefully so that we know $u_{\beta,n_\alpha} \notin G[s-1]$, and in fact, $u_{\beta,n_\alpha} \notin G[t]$ for all $t < s$. This allows α to enumerate the use u_{β,n_α} into $G[s]$ to destroy the f_β^G computation on the $2n_\alpha$th and $(2n_\alpha+1)$st components in $\mathcal{A}$.

We will refer to these two actions as α **issuing a challenge** to β. α now takes the outcome w_2. If by the next α-stage we have that α has not been initialized, then it takes the success outcome s. By α's actions above, we have that $\Phi_e(a_{2n}) = b_{2n}$ and $\Phi_e(a_{2n+1}) = b_{2n+1}$, and so Φ_e fails to be a computable isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

2.2.3 Being computably categorical relative to G

Since we want to make $\mathcal{A}$ computably categorical relative to our 1-generic G, we must define embeddings using G as an oracle. Additionally, we are looking at (partial) G-computable directed graphs $\mathcal{M}_i^G$ with domain ω whose edge relation is given by Φ_i^G. Any changes in initial segments of G can cause changes for both our partial G-computable embeddings and in $\mathcal{M}_i^G$ throughout the construction.

We define the following terms to keep track of certain finite subgraphs which appear and may remain in $\mathcal{M}_i^G$ throughout our construction.

Definition 2.2.1. Let C_0 and C_1 be isomorphic finite distinct subgraphs of $\mathcal{M}_i^G[s]$. The **age of $\mathbf{C_0}$** is the least stage $t \le s$ such that all edges in C_0 appear in $\mathcal{M}_i^G[t]$ and remained in $\mathcal{M}_i^G[s']$ for all $t \le s' \le s$, denoted by $\text{age}(C_0)$. We say that $\mathbf{C_0}$ is **older than $\mathbf{C_1}$** when $\text{age}(C_0) \le \text{age}(C_1)$.

We say that C_0 is the **oldest** if for all finite distinct subgraphs $C \cong C_0$ of $\mathcal{M}_i^G[s]$, $\text{age}(C_0) \le \text{age}(C)$.

Definition 2.2.2. Let $C_0 = \langle a_0, a_1, \ldots, a_k \rangle$ and $C_1 = \langle b_0, b_1, \ldots, b_k \rangle$ be isomorphic finite distinct subgraphs of $\mathcal{M}_i^G[s]$ with $a_0 < a_1 < \cdots < a_k$ and $b_0 < b_1 < \cdots < b_k$. We say that $C_0 <_{\text{lex}} C_1$ if for the least j such that $a_j \neq b_j$, $a_j < b_j$.

We say that C_0 is the **lexicographically least** if for all finite distinct subgraphs $C \cong C_0$ of $\mathcal{M}_i^G[s]$, $C_0 <_{\text{lex}} C$.

If $\mathcal{A} \cong \mathcal{M}_i^G$, then we need to build a G-computable isomorphism between these graphs. To achieve this, we meet the following requirement for each $i \in \omega$.

$$S_i : \text{if } \mathcal{A} \cong \mathcal{M}_i^G, \text{ then there exists a } G\text{-computable isomorphism } f_i^G : \mathcal{A} \to \mathcal{M}_i^G.$$

We will show in the verification that if $\mathcal{A} \cong \mathcal{M}_i^G$, "true" copies of components from $\mathcal{A}$ will eventually appear and remain in $\mathcal{M}_i^G$ (and thus become the oldest and lex-least finite subgraph which is isomorphic to a component in $\mathcal{A}$), and so our S_i-strategy below will be able to define the correct G-computable isomorphism between the two graphs.

Let α be an S_i-strategy. When α is first eligible to act, it sets its parameter $n_\alpha = 0$ and defines f_α^G to be the empty map. Once α has defined n_α, then when α acted at the previous α-stage s_0, we have the following situation:

- For each $m < n_\alpha$, $f_\alpha^G[s_0]$ maps the $2m$th and $(2m+1)$st components of $\mathcal{A}[s_0]$ to isomorphic copies in $\mathcal{M}_i^G[s_0]$.

- For $m < n_\alpha$, let l_m be the maximum $\Phi_i^G[s_0]$-use for the loops in the copies in $\mathcal{M}_i^G[s_0]$ for the $2m$th and $(2m+1)$st components in $\mathcal{A}$. We can assume that if $m_0 < m_1 < n_\alpha$, then $l_{m_0} < l_{m_1}$.

- For $m < n_\alpha$, let $u_{\alpha,m}$ be the $f_\alpha^G[s_0]$-use for the mapping of the $2m$th and $(2m+1)$st components of $\mathcal{A}$. This use will be constant for all elements in these components.

- By construction, we will have that $l_m < u_{\alpha,k}$ for all $m \leq k < n_\alpha$.

Suppose α is acting at stage s and that $n_\alpha > 0$ and let s_0 be the previous α-stage in the construction. We now break into cases.

If α took an outcome w_n at s_0, then no strategy could have changed G below the use $u_{\alpha,n_\alpha-1}$, and so the value of n_α remains unchanged.

If instead we have that α took the ∞ outcome at s_0, then G may have changed underneath $u_{\alpha,n_\alpha-1}$. We will show in the verification that the only strategies that can change G below this use are P or R-strategies β such that $\beta \supseteq \alpha^\frown\langle\infty\rangle$. If $G[s] \upharpoonright (u_{\alpha,n_\alpha-1}+1) \neq G[s_0] \upharpoonright (u_{\alpha,n_\alpha-1}+1)$, then some $\beta \supseteq \alpha^\frown\langle\infty\rangle$ caused a change at stage s_0 after α acted. Furthermore, at most one such β can cause this change at stage s_0, as every other strategy extending $\beta^\frown\langle s\rangle$ will define new and large witnesses for the remainder of stage s_0.

If β is a P-strategy which changed G under $u_{\alpha,n_\alpha-1}$, then it added diagonalizing loops to the $2n_\beta$th and $(2n_\beta+1)$st components in $\mathcal{A}$. Assuming that $n_\beta < n_\alpha$, then α has already defined $f_\alpha^G[s_0]$ on these components. Hence, β enumerates the use u_{α,n_β} into G to destroy this computation. To account for this change, α redefines n_α to be the least m such that f_α^G is not currently defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$.

If instead we have that the β is an R-strategy, then suppose β redefines G by setting $G[s_0] = \tau^\frown 0^\omega$. This definition may have changed G on numbers as small as n_α and hence may have injured previously defined f_α^G computations. So, at stage s, α resets n_α to be the least m such that f_α^G is not currently defined on the $2m$th and $(2m+1)$st components of $\mathcal{G}$.

We now carry out the main action of the S_i-strategy: we check whether we can extend $f_\alpha^G[s-1]$ to the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}[s]$. Search for isomorphic copies in $\mathcal{M}_i^G[s]$ of these components. If there are multiple copies in $\mathcal{M}_i^G[s]$, choose the oldest such copy to map to, and if there are multiple equally old copies, choose the lexicographically least oldest copy. If there are no copies in $\mathcal{M}_i^G[s]$, then keep the value of n_α the same, f_α^G unchanged, and let the next requirement act. Otherwise, extend $f_\alpha^G[s-1]$ to $f_\alpha^G[s]$ to include the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ with a large use u_{α,n_α}. Since u_{α,n_α} was chosen large, we have that $u_{\alpha,n_\alpha} > l_k$ for all $k \leq n_\alpha$. Increment n_α by 1 and go to the next requirement. If α had to redefine n_α because it was challenged by an S-strategy β of higher priority, then α continues to take finitary outcomes until the value of n_α exceeds n_β.

If $\mathcal{A} \cong \mathcal{M}_i^G$, then for each n, eventually the real copies of the $2n$th and $(2n+1)$st components of $\mathcal{A}$ will appear in $\mathcal{M}_i^G$. Moreover, they will eventually be the oldest and lexicographically least copies in $\mathcal{M}_i^G$. After G has settled down on the the maximum true G-use on the edges in the loops in these $\mathcal{M}_i^G$ components, we will define f_α^G correctly on these components with a large use $u_{\alpha,n}$. Therefore, eventually our map f_α^G is never injured again on the $2n$th and $(2n+1)$st components. It follows that if $A \cong \mathcal{M}_i^G$, then f_α^G will be an embedding of $\mathcal{A}$ into $\mathcal{M}_i^G$ which can be extended to a G-computable isomorphism defined on all of $\mathcal{A}$.

2.2.4 An interaction caused by genericity

The following interaction arises because we are building a 1-generic. The fact that initial segments of G can change several times throughout the construction requires our isolated strategies to be more flexible than what was done previously, and we outline the needed changes to affected strategies below.

Let α be an S_i-strategy, β be an R_j-strategy, and γ be a P_e-strategy where $\alpha^\frown\langle\infty\rangle \subseteq \beta^\frown\langle w_1\rangle \subseteq \gamma$, where $\beta^\frown\langle w_1\rangle$ indicates that β defined its parameter n_β and took the waiting outcome at the last β-stage. Suppose that n_β, at a stage s_0, and n_γ have been defined and that $n_\beta < n_\gamma$. Suppose that at stage s_1, α is able to define $f^G_\alpha[s_1]$ with a use u_{α,n_γ} on the $2n_\gamma$th and $(2n_\gamma+1)$st components of $\mathcal{A}$, with the configuration of the loops in $\mathcal{A}[s_1]$ and $\mathcal{M}^G_i[s_1]$ being:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1 \qquad c : 2, 5n_\gamma + 1$$

$$a_{2n_\gamma+1} : 2, 5n_\gamma + 2 \qquad d : 2, 5n_\gamma + 2.$$

Note that $f^G_\alpha[s_1]$ on these two $\mathcal{A}$-components is being protected by the initial segment $G[s_1] \upharpoonright (u_{\alpha,n_\gamma} + 1)$. Then, at the end of stage s_1, α takes the ∞ outcome, and β is eligible to act. Suppose that β continues to take the w_1 outcome, and so γ is eligible to act at a stage $s_2 > s_1$. Suppose that γ sees that the map $\Phi_e[s_2]$ maps its chosen $\mathcal{A}$-components isomorphically to their copies in $\mathcal{B}[s_2]$, and thus begins to diagonalize by adding new loops to the following $\mathcal{A}$-components and $\mathcal{B}$-components:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3 \qquad b_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4$$

$$a_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4 \qquad b_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3.$$

When γ adds these loops, it enumerates u_{α,n_γ} into $G[s_2]$, and so $G[s_2] \upharpoonright (u_{\alpha,n_\gamma} + 1) \neq G[s_1] \upharpoonright (u_{\alpha,n_\gamma} + 1)$ and $f^G_\alpha[s_1]$ disappears on the the $2n_\gamma$th and $(2n_\gamma + 1)$st components of $\mathcal{A}$. Let $s_3 \geq s_2$ be a stage such that newly added loops first appeared in $\mathcal{M}^G_i[s_3]$ in the following positions:

$$a_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3 \qquad c : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4$$

$$a_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 4 \qquad d : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3.$$

α is now able to recover its map f^G_α on these $\mathcal{A}$-components with a new large use u'_{α,n_γ} by mapping a_{2n_γ} to d, $a_{2n_\gamma+1}$ to c, and all nodes in each component to their respective copies in $\mathcal{M}^G_i[s_3]$. Note that this new map $f^G_\alpha[s_3]$ on these $\mathcal{A}$-components is being protected by $G[s_3] \upharpoonright (u'_{\alpha,n_\gamma} + 1)$. At the end of stage s_3, α takes the ∞ outcome and suppose β is now eligible to act again. Now, suppose that β has found an extension τ_{n_β} of $G[s_0] \upharpoonright n_\beta$ which is in $W_j[s_4]$ where $s_4 > s_3$. β then extends $G[s_0] \restriction n_\beta$ to τ_{n_β} and defines $G[s_4] = \tau_{n_\beta}^\frown 0^\omega$. Furthermore, suppose that $G[s_4] \restriction (u_{\alpha,n_\gamma}+1) = G[s_1] \restriction (u_{\alpha,n_\gamma}+1)$, and so the original map $f_\alpha^G[s_1]$ is restored on the $2n_\gamma$th and $(2n_\gamma+1)$st components of $\mathcal{A}$. β taking the success outcome initializes γ since $\gamma \supseteq \beta^\frown\langle w_1\rangle$, however because $\mathcal{A}$ and $\mathcal{B}$ need to be computable directed graphs, the loops added at stage s_2 must remain in both graphs. In particular, if the true outcome for $\mathcal{M}_i^G$ is to have the following connected components

$$c : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+4$$

$$d : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+3,$$

then because $f_\alpha^G[s_1]$ was restored on the two $\mathcal{A}$-components, it is now incorrect on the G-computable embedding from $\mathcal{A}$ into $\mathcal{M}_i^G$ since $f_\alpha^G[s_1](a_{2n_\gamma}) = c$ and $f_\alpha^G[s_1](a_{2n_\gamma+1}) = d$, in particular:

$$\begin{array}{ll} a_{2n_\gamma} : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+3 & c : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+4 \\ a_{2n_\gamma+1} : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+4 & d : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+3. \end{array}$$

We have then that if $\alpha^\frown\langle\infty\rangle$ is on the true path, it will define an incorrect a G-computable embedding from $\mathcal{A}$ into $\mathcal{M}_i^G$. To resolve this issue, after β finds its extension, we will add an additional step to the R_j-strategy where we homogenize any affected $\mathcal{A}$-components defined so far, namely the $2n_\gamma$th and $(2n_\gamma+1)st$ components of $\mathcal{A}$ in this example, in the following way:

$$a_{2n_\gamma} : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+3, 5n_\gamma+4$$

$$a_{2n_\gamma+1} : 2, 5n_\gamma+1, 5n_\gamma+2, 5n_\gamma+3, 5n_\gamma+4.$$

Then, if the newly added loops reappear in the corresponding copies of each component in $\mathcal{M}_i^G$ and remain, it does not matter which positions they appear in since $f_\alpha^G[s_1]$ will be correct no matter what. If other cycles or the loops never appear again after β's success, then we have that $\mathcal{A} \ncong \mathcal{M}_i^G$ and so α trivially wins. Furthermore, the path now goes to the left of γ, and so γ is initialized, and so we do not hurt our construction by undoing γ's diagonalization by homogenizing.

Additionally, since we need that $\mathcal{A} \cong \mathcal{B}$, we must also homogenize the corresponding components in $\mathcal{B}$ as well:

$$b_{2n_\gamma} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3, 5n_\gamma + 4$$

$$b_{2n_\gamma+1} : 2, 5n_\gamma + 1, 5n_\gamma + 2, 5n_\gamma + 3, 5n_\gamma + 4.$$

Lastly, we will also homogenize the $\mathcal{A}$-components which received diagonalizing loops from P-strategies which are to the right of the current true path.

2.3 Proof of Theorem 2.1.7

2.3.1 Requirements

We have three requirements for our construction:

$$R_j : (\exists \sigma \subseteq G)(\sigma \in W_j \vee (\forall \tau \supseteq \sigma)(\tau \notin W_j));$$

$$P_e : \Phi_e : \mathcal{A} \to \mathcal{B} \text{ is not an isomorphism;}$$

$$S_i : \text{if } \mathcal{A} \cong \mathcal{M}_i^G, \text{ then there exists a } G\text{-computable isomorphism } f_i^G : \mathcal{A} \to \mathcal{M}_i^G.$$

2.3.2 Construction

Let $\Lambda = \{\infty <_\Lambda \cdots <_\Lambda s <_\Lambda w_2 <_\Lambda w_1 <_\Lambda w_0\}$ be the set of outcomes, and let $T = \Lambda^{<\omega}$ be our tree of strategies. The construction will be performed in ω many stages s.

We define the **current true path** p_s, the longest strategy eligible to act at stage s, inductively. For every s, λ, the empty string, is eligible to act at stage s. Suppose the strategy α is eligible to act at stage s. If $|\alpha| < s$, then follow the action of α to choose a successor $\alpha^\frown\langle o \rangle$ on the current true path. If $|\alpha| = s$, then set $p_s = \alpha$. For all strategies β such that

$p_s <_L \beta$, initialize β (i.e., set all parameters associated to β to be undefined). Also, if $p_s <_L \beta$ and β is a P-strategy such that the $2n_\beta$th and $(2n_\beta + 1)$st components of $\mathcal{A}$ and $\mathcal{B}$ have been diagonalized (i.e., the $(5n+3)$- and $(5n+4)$-loops have been added), then homogenize these components in $\mathcal{A}$ and $\mathcal{B}$. If $\beta <_L p_s$ and $|\beta| < s$, then β retains the same values for its parameters.

We will now give formal descriptions of each strategy and their outcomes in the construction.

2.3.3 R_j-strategies and outcomes

We first cover the R_j-strategies used to build our 1-generic set G. Let α be an R_j-strategy eligible to act at stage s.

Case 1: If α is acting for the first time at stage s or has been initialized since the last α-stage, it defines $m_\alpha = \max\{n_\beta : n_\beta \text{ defined for } \beta \subset \alpha\}$ only when n_β has been defined for all $\beta \subset \alpha$. Once m_α has been defined, α waits to see if f^G_γ converges on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for all S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \alpha$. Until f^G_γ converges on all of those components, it remains in **Case 1** by taking the w_0 outcome and does not define its parameter n_α.

Case 2: If α took the w_0 outcome at the previous α-stage and all maps f^G_γ have converged on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \alpha$, it defines its parameter n_α to be large. In particular, n_α is greater than all of the uses of f^G_γ on the $2n_\beta$th and $(2n_\beta + 1)$st components for all P- and R-strategies $\beta \subset \alpha$ (since α won't define n_α until all the n_β's are defined). Then, take outcome w_1.

Case 3: If α took the w_1 outcome at the end of the previous α-stage, check if there is an extension τ of $G[s-1] \upharpoonright n_\alpha$ such that $\tau \in W_j[s]$. If not, take the w_1 outcome. If such a τ is found, define $G[s] = \tau^\frown 0^\omega$ and take outcome w_2.

Case 4: If α took the w_2 outcome the last time it was eligible to act and has not been initialized, take outcome s.

Case 5: If α took the s outcome the last time it was eligible to act and has not been initialized, continue taking outcome s.

2.3.4 P_e-strategies and outcomes

Let α be a P_e-strategy eligible to act at stage s.

Case 1: If α is first eligible to act at stage s or has been initialized, it defines $m_\alpha = \max\{n_\beta : n_\beta \text{ defined for } \beta \subset \alpha\}$ only when n_β has been defined for all $\beta \subset \alpha$. Once m_α has been defined, α waits to see if f_γ^G converges on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for all S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \alpha$. Until f_γ^G converges on all of those components, α remains in **Case 1** by taking the w_0 outcome and does not define define its parameter n_α.

Case 2: If α took the w_0 outcome at the previous α-stage and all maps f_γ^G have converged on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$ for S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \alpha$, it defines the parameter $n_\alpha = n$ to be large and takes outcome w_1.

Case 3: If α took the w_1 outcome at the last α-stage, check whether $\Phi_e[s]$ maps the $2n$th and $(2n+1)$st components of $\mathcal{A}$ isomorphically into $\mathcal{B}$. If not, take outcome w_1.

If so, add a $(5n+2)$- and $(5n+3)$-loop to a_{2n} and a $(5n+1)$- and $(5n+4)$-loop to a_{2n+1} in $\mathcal{A}[s]$. Add a $(5n+2)$- and $(5n+4)$-loop to b_{2n} and a $(5n+1)$- and $(5n+3)$-loop to b_{2n} in $\mathcal{B}[s]$. If a map f_β^G had already been defined on these components with a use $u_{\beta,n}$ by an S-strategy β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, enumerate $u_{\beta,n}$ into G, and issue a challenge to each such S-strategy. Take outcome w_2.

Case 4: If α took the w_2 outcome at the last α-stage and has not been initialized, take outcome s.

Case 5: If α took the s outcome at the previous α-stage and has not been initialized, take outcome s again.

2.3.5 S_i-strategies and outcomes

Let α be an S_i-strategy eligible to act at stage s.

Case 1: If α is acting for the first time or has been initialized since the last α-stage, set $n_\alpha = 0$, define $f_\alpha^G[s]$ to be the empty function, and take the w_0 outcome.

Case 2: If we are not in **Case 1**, let t be the last α-stage and we break into the following subcases.

Subcase 1: α is not currently challenged by a P-strategy and one of the following conditions holds:

- α took the w_{n_α} outcome at stage t,
- α took the ∞ outcome at stage t but no P or R-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$ took the w_1 outcome at t, or
- α took the ∞ outcome at t and an R-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$ took the w_1 outcome at t but $G[s-1] \upharpoonright (u_{\alpha,n_\alpha-1}+1) = G[t] \upharpoonright (u_{\alpha,n_\alpha-1}+1)$.

In this case, α passes to the module for checking for extensions of f_α^G below.

Subcase 2: α is currently challenged by a P-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$. In the verification, we will show that β is unique.

If α was challenged by β at stage t (i.e., s is the first α-stage since α was challenged), then set n_α to be the least m such that $f_\alpha^G[s-1]$ is not defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$. In the verification, we will show that $n_\alpha \leq n_\beta$. Go to the module for checking for extensions of f_α^G. If f_α^G is extended to the $2n_\alpha$th and $(2n_\alpha+1)$st components, then increment n_α. If $n_\alpha > n_\beta$, take the ∞ outcome and remove the challenge on α. Otherwise, take the w_{n_α} outcome and the challenge on α remains.

If α was challenged before stage t, then α acts in the same way except it skips the initial redefining of n_α.

Subcase 3: α is not currently challenged by a P-strategy, but α took the ∞ outcome at stage t, an R-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$ took outcome w_1 at stage t, and $G[s-1] \upharpoonright (u_{\alpha,n_\alpha-1}+1) \neq G[t] \upharpoonright (u_{\alpha,n_\alpha-1}+1)$. Reset n_α to be the least m such that $f_\alpha^G[s-1]$ is not defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$. Go to the module for checking for extensions of f_α^G.

Module for checking for extensions of f^G_α: α searches for the oldest and lex-least copies of the $2n_\alpha$th and $(2n_\alpha+1)$st components in $\mathcal{M}_i^G[s]$. If no copies are found, leave f^G_α and n_α unchanged and take outcome w_{n_α}. If α is challenged, then it remains challenged. Otherwise, if such copies are found, extend f^G_α by mapping the $2n_\alpha$th and $(2n_\alpha+1)$st components in $\mathcal{A}$ onto their copies in $\mathcal{M}_i^G[s]$. Define u_{α,n_α} large and set it as the use of the newly defined computations of f^G_α. Increment n_α by 1 and take the ∞ outcome (unless α is challenged and we still have that $n_\alpha \leq n_\beta$, in which case, take the w_{n_α} outcome).

2.3.6 Verification

We begin with proving several auxiliary lemmas based on key observations of the construction. Afterwards, we will prove the main verification lemma.

Lemma 2.3.1. If $f : \mathcal{A} \to \mathcal{A}$ is an embedding of $\mathcal{A}$ into itself, then f is an isomorphism.

Proof. The proof is largely the same as before. □

Lemma 2.3.2. If $\mathcal{A} \cong \mathcal{M}_i^G$ for a G-computable directed graph $\mathcal{M}_i^G$ and $f^G : \mathcal{A} \to \mathcal{M}_i^G$ is an embedding defined on all of $\mathcal{A}$, then f^G is an isomorphism.

Proof. This follows immediately from Lemma 2.3.1. □

Lemma 2.3.3. An S-strategy α will be challenged by at most one P-strategy at any given stage.

Proof. Let α be an S-strategy and let β be a P-strategy such that $\beta \supseteq \alpha^\frown\langle\infty\rangle$. If β challenges α at some stage s, then it takes the w_2 outcome for the first time. The P-strategies extending $\beta^\frown\langle w_2\rangle$ will then wait as in **Case 1**, and so will not challenge α. Until α can meet its challenge, it will continue to take the w_{n_β} outcome, and so P-strategies extending $\alpha^\frown\langle w_{n_\beta}\rangle$ will not be able to challenge α since $w_{n_\beta} \neq \infty$. □

The next two lemmas will prove facts about our 1-generic G that will be useful for our S-strategies.

Lemma 2.3.4. G is Δ_2^0.

Proof. Let m be arbitrary and we will show that G can only change finitely often below m. The only strategies which can change G are the R- and P-strategies. If α is a P-strategy, then α changes G below m if and only if its parameter $n_\alpha \leq m$. If α is instead an R-strategy, then α changes G below m if and only if there is an S-strategy $\beta^\frown\langle\infty\rangle \subseteq \alpha$ such that $u_{\beta,n_\alpha} \leq m$. In either case, α will only act once to change G according to its strategy unless it is initialized and chooses new large parameters. Since parameters are always chosen large (and so are never reused), only finitely many strategies can change G below m. □

Lemma 2.3.5. If $\mathcal{A} \cong \mathcal{M}_i^G$, then for each n, there is an s such that for all $t \geq s$, the true copies of the $2n$th and $(2n+1)$st components of $\mathcal{A}$ in $\mathcal{M}_i^G$ are the oldest and lexicographically least isomorphic copies in $\mathcal{M}_i^G[t]$ of these components.

Proof. By Lemma 2.3.4, we have that G is Δ_2^0. If $\mathcal{A} \cong \mathcal{M}_i^G$, then for each n, we have true copies of the $2n$th and $(2n+1)$st components of $\mathcal{A}$ in $\mathcal{M}_i^G$. Suppose that the associated G-uses for the true copies of the $2n$th and $(2n+1)$st components are u_n and u_{n+1}, respectively. Since G is Δ_2^0, there exists a sufficiently large stage s_n such that $G[s_n] \upharpoonright u_n = G \upharpoonright u_n$, and so after stage s_n, the copy of the $2n$th component in $\mathcal{A}$ will become the oldest and lexicographically least isomorphic copy of the component in $\mathcal{M}_i^G$. The same holds true for the $(2n+1)$st component of $\mathcal{A}$. □

Lemmas for each strategy

For each strategy in our construction, we prove several key lemmas. We begin with the P- and R-strategies.

Lemma 2.3.6. Let α be a P- or R-strategy. If β is another P- or R-strategy such that $\beta^\frown\langle w_0\rangle \subseteq \alpha$, then α takes the w_0 outcome at every α-stage.

Proof. If $\beta^\frown\langle w_0\rangle \subseteq \alpha$ and s is an α-stage, then n_β is not defined at stage s. Therefore, α doesn't define n_α at stage s, and so it takes the w_0 outcome. □

Once an R-strategy α defines its parameter, it will stay defined for the rest of the construction unless α is initialized. We show that if α is on the true path, then it will define an initial segment of G that either meets or avoids its given c.e. set. We first begin with a short lemma.

Lemma 2.3.7. Let α be an R- or P-strategy that acts in **Case 3** at stage s. Then $G[s] \upharpoonright n_\alpha = G[s-1] \upharpoonright n_\alpha$.

Proof. If α is an R-strategy, this follows immediately because $G[s-1] \upharpoonright n_\alpha \subseteq \tau$. If α is a P-strategy, then α may enumerate a use u_{β,n_α} into G where β is an S-strategy where $\beta^\frown\langle\infty\rangle \subseteq \alpha$. In this case, since u_{β,n_α} was chosen large, we have that $u_{\beta,n_\alpha} > n_\alpha$ and so the only change to G occurs above n_α. □

Lemma 2.3.8. Let α be an R_j-strategy that defines n_α at a stage s_0.

(1) Unless α is initialized, at all stages $s > s_0$, we have that $G[s] \upharpoonright n_\alpha = G[s_0] \upharpoonright n_\alpha$.

(2) Suppose that α acts as in **Case 3** at a stage $s_1 > s_0$ with the string τ where $\tau \in W_j[s_1]$. Unless α is initialized, for all stages $s > s_1$, we have that $G[s] \upharpoonright |\tau| = \tau$.

Proof. Let α and n_α be as above. For (1), by Lemma 2.3.7 all R- or P-strategies β where $\beta \supset \alpha$ cannot change $G[s]$ below n_α, and hence cannot change G below n_α. Therefore, if there is a change to $G[s]$ below n_α, it was caused by some R- or P-strategy δ acting in **Case 3** with $\delta^\frown\langle w_1\rangle \subseteq \alpha$. But when δ acts, it takes the w_2 outcome, which initializes α.

For (2), we have that α takes the w_2 outcome at stage s_1 and the s outcome after stage s_1. All $\beta \supseteq \alpha^\frown\langle w_2\rangle$ or $\beta \supseteq \alpha^\frown\langle s\rangle$ will define large parameters $n_\beta > |\tau|$. So if there are any changes to G below $|\tau|$, it is caused by some R- or P-strategy $\delta^\frown\langle w_1\rangle \subseteq \alpha$. However, if δ changes G, then it takes the w_2 outcome, but this initializes α. □

We can also prove something similar for the S-strategies.

Lemma 2.3.9. Let α be an S-strategy and let $s_0 < s_1$ be α-stages such that $n_\alpha[s]$ has already been defined and α is not initialized between s_0 and s_1. Then

$$G[s_0] \upharpoonright n_\alpha[s_0] = G[s_1] \upharpoonright n_\alpha[s_0]$$

unless some R- or P-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$ with $n_\beta < n_\alpha[s_0]$ acts between stages s_0 and s_1.

Proof. By Lemma 2.3.8, G can change below $n_\alpha[s_0]$ only if an R- or P-strategy β with $n_\beta < n_\alpha[s_0]$ acts. Therefore, we cannot have that $\alpha^\frown\langle\infty\rangle <_L \beta$. We also cannot have that $\beta <_L \alpha$ or $\beta \subseteq \alpha$ since α would get initialized. Hence, it must be that $\alpha^\frown\langle\infty\rangle \subseteq \beta$. $\square$

We now prove that if a P_e-strategy α takes action to add diagonalizing loops to a pair of A-components, then the diagonalization against Φ_e on these components will remain at the end of the construction.

Lemma 2.3.10. Let α be a P-strategy. Suppose that α adds diagonalizing loops to the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ at a stage s_0. Unless α is initialized, these components are not homogenized at a future stage.

Proof. Let α be as in the statement of the lemma. We only homogenized these components if the path moves to the left of α, in which case α is initialized. $\square$

We now prove several key facts for the S-strategies that will help with the main verification lemma later on.

Lemma 2.3.11. Let α be an S-strategy and β be a P-strategy such that $\alpha^\frown\langle\infty\rangle \subseteq \beta$. Suppose that there are stages $t_0 < t_1 < t_2 < t_3$ such that β defines m_β at t_0, β defines n_β at t_1, β acts in **Case 3** at t_2, t_3 is the first α-stage after t_2, and β is not initialized between t_0 and t_3. Then,

(1) $n_\alpha[t_1] < n_\beta[t_1]$, and for all S-strategies γ such that $\gamma^\frown\langle\infty\rangle \subseteq \beta$, f^G_γ is defined on the $2m_\beta$th and $(2m_\beta+1)$st components of $\mathcal{A}$ with $n_\beta[t_1] > u_{\gamma,m_\beta}[t_1]$;

(2) for $t_1 \leq t < t_2$, $n_\alpha[t] > m_\beta[t] = m_\beta[t_0]$; and

(3) $n_\alpha[t_3] > m_\beta[t_3] = m_\beta[t_0]$.

Proof. Note that the values of m_β and n_β do not change once they are defined since β is not initialized. For (1), β defines n_β large at stage t_1, and so $n_\beta[t_1] > n_\alpha[t_1]$. In addition, β does not define n_β until all S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \beta$ have $n_\gamma > m_\beta$. Therefore, when n_β is chosen large at stage t_1, we have that $n_\beta[t_1] > u_{\gamma,m_\beta}[t_1]$.

For (2), when β defines n_β at stage t_1, f_α^G is defined on the $2m_\beta$th and $(2m_\beta+1)$st components of $\mathcal{A}$. Therefore, we have that $n_\alpha[t_1] > m_\beta[t_0]$. At stage t_1, β takes the w_1 outcome for the first time, and so all strategies γ where $\beta^\frown\langle w_1\rangle \subseteq \gamma$ choose parameters larger than $n_\alpha[t_1]$. In particular, none of these strategies can lower the value of n_α below m_β. The only other strategies which can lower n_α without initializing α are γ where $\alpha^\frown\langle\infty\rangle \subseteq \gamma \subset \beta$. However, these γ would initialize β. Hence, we have that for all $t_1 \leq t < t_2$ that $n_\alpha[t] > m_\beta$. In addition, at stage t_2, β acts after α, so n_α does not get redefined to reflect β's action until the α-stage t_3. It follows that $n_\alpha[t] > m_\beta$ for $t_2 \leq t < t_3$.

For (3), at stage t_3, α sets $n_\alpha[t_3]$ to be the least m such that $f_\alpha^G[t_3]$ is not defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$. At stage t_2, β enumerated u_{γ,n_β} for S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \beta$. However, $u_{\gamma,n_\beta} > n_\beta$ because u_{γ,n_β} was defined large when it was chosen and $n_\beta > u_{\alpha,m_\beta}$ by (1). Therefore, the computation f_α^G on the $2m_\beta$th and $(2m_\beta+1)$st components is not destroyed by β's action at stage t_2. It follows that $n_\alpha[t_3] > m_\beta[t_3] = m_\beta[t_0]$. $\square$

Lemma 2.3.12. Let α be an S-strategy and β be an R-strategy such that $\alpha^\frown\langle\infty\rangle \subseteq \beta$. Suppose that there are stages $t_0 < t_1 < t_2 < t_3$ such that β defines m_β at t_0, β defines n_β at t_1, β acts in **Case 3** at t_2, t_3 is the next α-stage after t_2, and β is not initialized between t_0 and t_3. Then,

(1) $m_\beta < n_\alpha[t_1] < n_\beta[t_1]$ and $n_\beta[t_1] > u_{\alpha,m_\beta}[t_1]$, which is the use of $f_\alpha^G[t_1]$ on the $2m_\beta$th and $(2m_\beta+1)$st components of $\mathcal{A}$;

(2) for all $t_1 \leq t < t_3$, $n_\alpha[t] > m_\beta$; and

(3) $n_\alpha[t_3] > m_\beta$.

Proof. (1) and (2) hold as in Lemma 2.3.11. For (3), when β acts at stage t_2, it defines $G[t_2] = \tau^\frown 0^\omega$ where $G[t_2 - 1] \upharpoonright n_\beta \subseteq \tau$. Therefore, if $u_{\alpha,m_\beta}[t_1]$ is the use of the map $f^G_\alpha[t_1]$ on the $2m_\beta$th and $(2m_\beta + 1)$st components, then

$$G[t_2] \upharpoonright u_{\alpha,m_\beta} = G[t_2 - 1] \upharpoonright u_{\alpha,m_\beta}.$$

Moreover, no requirements can change G below n_β between stages t_2 and t_3 without initializing β. Hence,

$$G[t_3 - 1] \upharpoonright u_{\alpha,m_\beta} = G[t_2 - 1] \upharpoonright u_{\alpha,m_\beta}.$$

In particular, when α acts in **Case 2** at stage t_3, either α acts in **Subcase 1** and doesn't change n_α, or it acts in **Subcase 3** and we retain $n_\alpha[s_3] > m_\beta$ because G has not changed below u_{α,m_β}. □

We can now combine Lemmas 2.3.11 and 2.3.12 into the following single lemma.

Lemma 2.3.13. Let α be an S-strategy. An R- or P-strategy $\beta \supseteq \alpha^\frown\langle\infty\rangle$ cannot cause n_α to drop below m_β.

With the help of Lemma 2.3.13, we now prove the following important fact about the S-strategies.

Lemma 2.3.14. Let α be an S-strategy that is never initialized after stage s and takes the ∞ outcome infinitely often. For all m, there exists a stage t_m such that $n_\alpha[t] \geq m$ for all $t \geq t_m$.

Proof. Fix m. Let $\beta_0, \ldots, \beta_k$ be the R- and P-strategies such that $\alpha^\frown\langle\infty\rangle \subseteq \beta_i$ and m_{β_i} is defined at some stage where $m_{\beta_i} < m$. Let $t > s$ be a stage such that no β_i acts as in **Case 3** to change G with $m_{\beta_i} < m$ after stage t. After stage t, no requirement can cause n_α to drop below m. Moreover, no other R- or P-requirement δ will act until $n_\alpha > m_\delta > m$. Let $t_m > t$ be the least stage such that $n_\alpha[t_m] > m$. □

Lemmas on interactions between multiple strategies

In this section, we prove several lemmas that detail how tension between multiple strategies are resolved in our construction. We first show that for a lower priority R-strategy, it won't be able to injure higher priority S-strategies at arbitrarily small numbers.

Lemma 2.3.15. Let α be an R_j-strategy that acts in **Case 3** at stage s. For all P- and R-strategies $\beta \subset \alpha$ and all S-strategies γ such that $\gamma^\frown\langle\infty\rangle \subseteq \alpha$, if f_γ^G is defined on the $2n_\beta$th and $(2n_\beta + 1)$st components of $\mathcal{A}$, then

$$G[s] \upharpoonright u_{\gamma,n_\beta} = G[s-1] \upharpoonright u_{\gamma,n_\beta}.$$

Therefore, these maps remain defined.

Proof. When α first acts, it waits for n_β to be defined for all $\beta \subset \alpha$ and then defines $m_\alpha = \max\{n_\beta : \beta \subset \alpha\}$ and does not define n_α until it sees that for all S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \alpha$, the map f_γ^G has been defined on the $2m_\alpha$th and $(2m_\alpha + 1)$st components of $\mathcal{A}$. When α is finally able to define n_α, it picks n_α large and so $n_\alpha > u_{\gamma,n_\beta}$ for all S-strategies $\gamma^\frown\langle\infty\rangle \subseteq \alpha$ and for all P- and R-strategies $\beta \subset \alpha$. If α finds a τ such that $\tau \in W_j[s]$ and $G[s-1] \upharpoonright n_\alpha \subseteq \tau$ at a stage s, it sets $G[s] = \tau^\frown 0^\omega$. This will not change G below u_{γ,n_β} for all γ and β as above since $n_\alpha > u_{\gamma,n_\beta}$. □

The following lemma details how the S-strategies are able to undo any of their previously defined maps on $\mathcal{A}$-components in the event that a pair of components were used to diagonalize against a computable Φ_e by some P_e-strategy.

Lemma 2.3.16. Let α be an S_i-strategy. If $f_\alpha^G[s]$ is defined on the $2m$th and $(2m+1)$st components of $\mathcal{A}$ at stage s, then $l_m[s] < u_{\alpha,m}[s]$ where $l_m[s]$ is the max use of the edges in the image of these components in $\mathcal{M}_i^G$ and $u_{\alpha,m}[s]$ is the use of this computation. In particular, any change in these loops in $\mathcal{M}_i^G$ causes the computation to be destroyed.

Proof. Whenever an $f_\alpha^G[s]$ computation is defined, its use is defined large. Therefore, we have that $l_m[s] < u_{\alpha,m}[s]$ when the computation is first defined. The computation may be destroyed later by a change in G below $u_{\alpha,m}[s]$. However, if it is later restored by $u_{\alpha,m}[t] = u_{\alpha,m}[s]$ for $t > s$, then $l_m[t] = l_m[s]$ and so the loops in $\mathcal{M}_i^G$ are restored as well. □

We now show that enumerating these uses has the intended effect of deleting a map f_α^G defined previously by some S-strategy α on some $\mathcal{A}$-components.

Lemma 2.3.17. Let α be a P-strategy and β be an S-strategy such that $\beta^\frown\langle\infty\rangle \subseteq \alpha$. Suppose $s_0 < s_1 < s_2$ are stages such that α defines n_α at s_0, β defined f_β^G on the $2n_\alpha$th and $(2n_\alpha + 1)$st components in $\mathcal{A}$ at s_1, α acts in **Case 3** at s_2 and puts u_{β,n_α} into $G[s_2]$, and α is not initialized between s_0 and s_2. Then,

(1) $n_\alpha[s_0] > n_\beta[s_0]$,

(2) for all $s_1 \leq t < s_2$, we have that the values of $n_\alpha[t] = n_\alpha[s_0]$ and $u_{\beta,n_\alpha}[t] = u_{\beta,n_\alpha}[s_1]$, so we denote them by n_α and u_{β,n_α},

(3) for all $t < s_2$, $u_{\beta,n_\alpha} \notin G[t]$ and therefore $G[s_2] \upharpoonright (u_{\beta,n_\alpha} + 1) \neq G[t] \upharpoonright (u_{\beta,n_\alpha} + 1)$ for all $t < s_2$,

(4) unless α is initialized, $u_{\beta,n_\alpha} \in G[s]$ for all $s \geq s_2$, and

(5) if there is an $s \geq s_2$ such that $u_{\beta,n_\alpha} \notin G[s]$, then the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{A}$ and $\mathcal{B}$ are homogenized.

Proof. (1) follows from the fact that n_α is defined large at stage s_0. For (2), n_α changes values only when α is initialized, and the only strategies that can change G below u_{β,n_α} would initialize α. For (3), when β defines u_{β,n_α} at stage s_1, the use was chosen large so $u_{\beta,n_\alpha} \notin G[t]$ for $t \leq s_1$. For $s_1 < t < s_2$, the only strategies that could change G below u_{β,n_α} would initialize α by Lemma 2.3.15. Lastly for (4), only an R-strategy γ could remove u_{β,n_α} and only if $n_\gamma < u_{\beta,n_\alpha}$. But such an R-strategy with $n_\gamma < u_{\beta,n_\alpha}$ would initialize α when it acts by Lemma 2.3.15. Furthermore, the path $p_s <_L \alpha$ causes the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{A}$ and $\mathcal{B}$ to be homogenized, proving (5). □

Lemma 2.3.18. Let α be a P-strategy and β be an S-strategy such that $\beta^\frown\langle\infty\rangle \subseteq \alpha$. Suppose α acts at stage t in **Case 3** and puts u_{β,n_α} into G and challenges β. Unless β is initialized, at the next β-stage s, β defines $n_\beta[s] \leq n_\alpha[s] = n_\alpha[t]$.

Proof. By Lemma 2.3.17 and the fact that β is not initialized, we get that $u_{\beta,n_\alpha} \in G[s-1]$. Therefore, $f^G_\beta[s-1]$ is no longer defined on the $2n_\alpha$th and $(2n_\alpha+1)$st components of $\mathcal{A}$ and hence n_β is redefined to a value which is at most n_α. □

Lemma 2.3.19 (Main Verification Lemma)**.** Let $TP = \liminf_s p_s$ be the true path of the construction, where p_s denotes the current true path at stage s of the construction. Let $\alpha \subset TP$.

(1) If α is an R_j-strategy, then the parameters m_α and n_α are eventually permanently defined and there is an outcome o and an α-stage t_α such that for all α-stages $s \geq t_\alpha$, α takes outcome o where o ranges over $\{s, w_1\}$.

(2) Let α be a P_e-strategy, then the parameters m_α and n_α are eventually permanently defined and there is an outcome o and an α-stage t_α such that for all α-stages $s \geq t_\alpha$, α takes outcome o where o ranges over $\{s, w_1\}$.

(3) If α is an S_i-strategy, then either α takes outcome ∞ infinitely often or there is an outcome w_n and a stage t_α such that for all α-stages $s > t_\alpha$, α takes outcome w_n. If $\mathcal{A} \cong \mathcal{M}^G_i$, then α takes the ∞ outcome infinitely often and defines an embedding $f^G_\alpha : \mathcal{A} \to \mathcal{M}^G_i$ which can be extended in a G-computable way to a G-computable isomorphism $\hat{f}^G_\alpha$ between $\mathcal{A}$ and $\mathcal{M}^G_i$.

In addition, α satisfies its assigned requirement.

Proof. For (1), let $\alpha \subset TP$ be an R_j-strategy and let s_0 be the least stage such that $\alpha \leq_L p_s$ for all $s \geq s_0$ and for all R- and P-strategies $\beta \subset \alpha$, n_β is defined. At stage s_0, α defines m_α permanently since α is never initialized again. By Lemma 2.3.14, there is an $s_1 > s_0$ such that for all S-strategies β where $\beta^\frown\langle\infty\rangle \subseteq \alpha$, we have $n_\beta[t] > m_\alpha$ for all $t \geq s_1$. At stage s_1 (if not before), α defines n_α permanently and takes outcome w_1.

If α remains in the first part of **Case 3** from its description for all α-stages $s \geq s_0$, then R_j is satisfied because for all extensions τ of $\sigma = G[s_0] \upharpoonright n_\alpha$, $\tau \notin W_j$. Moreover, since $\alpha \subset TP$, we have that $G[s_0] \upharpoonright n_\alpha = G \upharpoonright n_\alpha$ by Lemma 2.3.8. Otherwise, there is a stage $s \geq s_0$ such that α finds an extension $\tau \supseteq G[s_0] \upharpoonright n_\alpha$ where $\tau \in W_j[s]$. At the end of stage s, α takes the w_2 outcome, and since $\alpha \subset TP$, α will be able to act again and finally take the s outcome. For the rest of the construction, it will be in **Case 5** of its description after defining $\tau \subseteq G$ where $\tau \in W_j$. Again by Lemma 2.3.8, we have that $G[t] \upharpoonright |\tau| = \tau$ for all $t \geq s$, and R_j is satisfied.

We now prove (2). Suppose $\alpha \subset TP$ is a P_e-strategy. Let s_0 be the least stage such that $\alpha \leq_L p_s$ for all $s \geq s_0$ and for all R- or P-strategies $\beta \subset \alpha$, n_β is defined. Like before, at stage s_0, α defines m_α permanently since α is never initialized. By Lemma 2.3.14, there exists a stage $s_1 > s_0$ such that for all S-strategies β with $\beta^\frown\langle\infty\rangle \subseteq \alpha$, we have that $n_\beta[t] > m_\alpha$ for all $t \geq s_1$. Then at stage s_1 (if not before), α defines n_α permanently and takes outcome w_1.

If α remains in the first part of **Case 3** from its description for all α-stages $s \geq s_0$, then P_e is trivially satisfied since Φ_e is not total or maps $\mathcal{A}$ incorrectly into $\mathcal{B}$, and so it cannot be an isomorphism between $\mathcal{A}$ and $\mathcal{B}$. In this case, α takes outcome w_1 cofinitely often.

Otherwise, there is an α-stage $s_1 > s_0$ where $\Phi_e[s_1]$ maps the $2n$th and $(2n+1)$st components of $\mathcal{A}$ isomorphically into $\mathcal{B}$. Then, α carries out all actions described in the second part of **Case 3**. Let $s_2 > s_1$ be the next α-stage, and now α is in **Case 4** of its description and can now take the s outcome. Moreover, since $\alpha \subset TP$, α will never get initialized again and so the $2n$th and $(2n+1)$st components of $\mathcal{A}$ and $\mathcal{B}$ are never homogenized. By Lemma 2.3.10 we have that $\Phi_e(a_{2n}) = b_{2n}$, but a_{2n} is connected to a cycle of length $5n+3$ whereas b_{2n} is connected to a cycle of length $5n+4$. Similarly, $\Phi_e(a_{2n+1}) = b_{2n+1}$, but a_{2n+1} is connected to a cycle of length $5n+4$ whereas b_{2n+1} is connected to a cycle of length $5n+3$. Hence, Φ_e cannot be an isomorphism between $\mathcal{A}$ and $\mathcal{B}$, and so P_e is satisfied.

For (3), let $\alpha \subset TP$ be an S_i-strategy and let s_0 be the least stage such that for all stages $s \geq s_0$, $\alpha \leq_L p_s$. Under all of the subcases of **Case 3**, α enacts the module to search for extensions of f_α^G on the $2n_\alpha$th and $(2n_\alpha + 1)$st components of $\mathcal{A}$ for its currently defined n_α parameter. If an extension cannot be found for these components, then we have that α takes the w_{n_α} outcome at cofinitely many α-stages after s_0 and we are done since $\mathcal{A} \ncong \mathcal{M}_i^G$.

We can then assume that $\mathcal{A} \cong \mathcal{M}_i^G$ and that α takes the ∞ outcome infinitely often. α can then eventually find a correct extension on these components by Lemma 2.3.5. Additionally, we have by Lemma 2.3.14 that $n_\alpha[s] \to \infty$ as $s \to \infty$ where $n_\alpha[s]$ denotes the value of n_α at the end of stage s. We also have that by Lemmas 2.3.16 and 2.3.18, these embeddings will remain or be able to recover if they are challenged by lower priority strategies.

Although f_α^G is not defined on the homogenizing loops added at the end of each stage, we can G-computably extend f_α^G to a map $\hat{f}_\alpha^G$ defined on all of $\mathcal{A}$ in the following way. Since $\mathcal{A} \cong \mathcal{M}_i^G$, then these loops added at the end of each stage will eventually have true copies in the images under f_α^G on the affected components, and they will become the oldest and lex-least such copies by Lemma 2.3.5. We can find copies of these new loops by using G since $\mathcal{M}_i^G$ is G-computable, and once we know when these loops appear, we can extend f_α^G appropriately on these new components to obtain $\hat{f}_\alpha^G$. Our new map $\hat{f}_\alpha^G$ is still G-computable, and by Lemma 2.3.2, $\hat{f}_\alpha^G$ is our G-computable isomorphism between $\mathcal{A}$ and $\mathcal{M}_i^G$. $\square$

2.4 Other classes of structures

For this section, we restate the main result from the first chapter of this thesis.

Theorem 2.4.1. Let $P = (P, \leq)$ be a computable partially ordered set and let $P = P_0 \sqcup P_1$ be a computable partition. Then, there exists a computable computably categorical directed graph $\mathcal{G}$ and an embedding h of P into the c.e. degrees where $\mathcal{G}$ is computably categorical relative to each degree in $h(P_0)$ and is not computably categorical relative to each degree in $h(P_1)$.

We now show that the structure which witnesses the above behavior need not be a directed graph.

Theorem 2.4.2. Let $P = (P, \leq)$ be a computable partially ordered set and let $P = P_0 \sqcup P_1$ be a computable partition. For the following classes of structures, there exists a computable example in each class which satisfies the conclusion of Theorem 2.4.1: symmetric, irreflexive graphs; partial orderings; lattices; rings with zero-divisors; integral domains of arbitrary characteristic; commutative semigroups; and 2-step nilpotent groups.

We will use the codings given in [19] in order to code the directed graph $\mathcal{G}$ (and presentations of it) in the statement of Theorem 2.4.1 into a structure in one of the above classes of structures.

2.4.1 Codings

Suppose an abstract graph $\mathcal{G}$ and a structure $\mathcal{A}$, from one of the listed classes in the statement of Theorem 2.4.2, have computable presentations. Let G and A be particular copies of $\mathcal{G}$ and $\mathcal{A}$, respectively. We first state the following definitions before defining the coding from [19].

Definition 2.4.3. A relation U on a structure $\mathcal{A}$ is **invariant** if for every automorphism $f : \mathcal{A} \cong \mathcal{A}$, we have that $f(U) = U$.

Here, an n-ary relation U on a structure $\mathcal{A}$ is some subset of $|\mathcal{A}|^n$ where $|\mathcal{A}|$ denotes the underlying domain of $\mathcal{A}$.

Definition 2.4.4. A relation U on the domain of a structure $\mathcal{A}$ is **intrinsically computable** if for any computable $\mathcal{B}$ and computable isomorphism $f : \mathcal{A} \to \mathcal{B}$, the image $f(U)$ is computable.

Definition 2.4.5. Let $\mathbf{d}$ be a degree. A **d-computable defining family** for a structure $\mathcal{A}$ is a $\mathbf{d}$-computable set of existential formulas $\varphi_0(\vec{a}, x), \varphi_1(\vec{a}, x), \ldots$ such that $\vec{a}$ is a tuple of elements of $|\mathcal{A}|$, each $x \in |\mathcal{A}|$ satisfies some φ_i, and no two elements of $|\mathcal{A}|$ satisfy the same φ_i.

The main idea of the coding methods given in [19] is that there are intrinsically computable, invariant relations $D(x)$ and $R(x, y)$ on the domain of $\mathcal{A}$ such that taking the elements in $|\mathcal{A}|$ satisfying $D(x)$ and adding the relation on them defined by $R(x, y)$ gives a copy of the graph $\mathcal{G}$. This gives us a map from copies of the structure $\mathcal{A}$ to copies of $\mathcal{G}$, which we will write as $A \mapsto G_A$. In addition, there is a uniform computable functional taking copies of $\mathcal{G}$ to copies of $\mathcal{A}$, which we will write as $G \mapsto A_G$. Note that these maps can be applied repeatedly. For example, we can apply a map to go from A to G_A and then to A_{G_A}, a copy of the structure $\mathcal{A}$. Note that A_{G_A} and A are isomorphic as both are copies of the structure $\mathcal{A}$, but they are *not* the same presentation.

The coding methods in [19] satisfy the following list of properties:

(P0) For every presentation G of $\mathcal{G}$, the structure A_G is $\deg(G)$-computable.

(P1) For every presentation G of $\mathcal{G}$, there is a $\deg(G)$-computable map $g_G : |A_G| \to G$ such that $R^{A_G}(x, y) \iff E^G(g_G(x), g_G(y))$ for all $x, y \in |A_G|$.

(P2) If $f : |\mathcal{A}| \to |\mathcal{A}|$ is 1-to-1 and onto and $R(x, y) \iff R(f(x), f(y))$ for all $x, y \in |\mathcal{A}|$, then f can be extended to an automorphism of $\mathcal{A}$.

(P3) For every presentation G of $\mathcal{G}$, there is a $\deg(G)$-computable set of existential formulas $\varphi_0(\vec{a}, \vec{b}_0, x), \varphi_1(\vec{a}, \vec{b}_1, x), \ldots$ such that $\vec{a}$ is a tuple of elements from the universe of A_G, each $\vec{b}_i$ is a tuple of elements of $|A_G|$, each x in the universe of A_G satisfies some φ_i, and no two elements of the universe of A_G satisfy the same φ_i.

Each of these properties are key in proving Lemmas 2.6-2.9 in [19], whose relativized versions below are needed to prove Theorem 2.4.2. Here, we let $\mathcal{G}$ be the directed graph that satisfies the conclusion of Theorem 2.4.1, but the lemmas hold in general for any directed graph and for all degrees $\mathbf{d}$.

Lemma 2.4.6. For every $\mathbf{d}$-computable presentation G of $\mathcal{G}$, there is a $\mathbf{d}$-computable isomorphism from G_{A_G} onto G.

Lemma 2.4.7. For every **d**-computable presentation A of $\mathcal{A}$, there is a **d**-computable isomorphism from A_{G_A} onto A.

Lemma 2.4.8. If G_0 and G_1 are **d**-computable presentations of $\mathcal{G}$ and $h : G_0 \to G_1$ is a **d**-computable isomorphism, then there is a **d**-computable isomorphism $\hat{h} : A_{G_0} \to A_{G_1}$.

Lemma 2.4.9. If A_0 and A_1 are **d**-computable presentations of $\mathcal{A}$ and $h : A_0 \to A_1$ is a **d**-computable isomorphism, then there is a **d**-computable isomorphism $\hat{h} : G_{A_0} \to G_{A_1}$.

We now prove a key lemma with the help of these relativized lemmas.

Lemma 2.4.10. For any degree **d**, $\mathcal{G}$ is computably categorical relative to **d** if and only if $\mathcal{A}$ is computably categorical relative to **d**.

Proof. Fix the degree **d**. Suppose $\mathcal{G}$ is computably categorical relative to **d**. Let A_0 and A_1 be **d**-computable presentations of $\mathcal{A}$. We then obtain **d**-computable presentations G_{A_0} and G_{A_1} of $\mathcal{G}$. Since $\mathcal{G}$ is computably categorical relative to **d**, there is a **d**-computable isomorphism $h : G_{A_0} \to G_{A_1}$. By Lemma 2.4.8, we have a **d**-computable isomorphism $\hat{h} : A_{G_{A_0}} \to A_{G_{A_1}}$. Additionally, by Lemma 2.4.7, there are **d**-computable isomorphisms $f_0 : A_{G_{A_0}} \to A_0$ and $f_1 : A_{G_{A_1}} \to A_1$. It follows that $f_1 \circ \hat{h} \circ f_0^{-1} : A_0 \to A_1$ is a **d**-computable isomorphism as needed.

For the reverse direction, suppose $\mathcal{A}$ is computably categorical relative to **d**. To show $\mathcal{G}$ is computably categorical relative to **d**, we run an identical argument as above using Lemmas 2.4.6 and 2.4.9 in place of Lemmas 2.4.7 and 2.4.8, respectively. $\square$

Theorem 2.4.2 follows almost immediately, as we will now show.

Proof of Theorem 2.4.2. Let P be our computable poset where $P = P_0 \sqcup P_1$ is a computable partition, let $\mathcal{G}$ be the computable directed graph which witnesses Theorem 2.4.1, and let h be the embedding of P into the c.e. degrees. By Lemma 2.4.10, we have that since $\mathcal{G}$ is computably categorical, so is $\mathcal{A}$. We also have that because $\mathcal{G}$ is computably categorical relative to all $\mathbf{d} \in h(P_0)$ that $\mathcal{A}$ is computably categorical relative to all $\mathbf{d} \in h(P_0)$. Finally, since $\mathcal{G}$ is not computably categorical relative to any $\mathbf{d} \in h(P_1)$, it follows that $\mathcal{A}$ is also not computably categorical relative to any $\mathbf{d} \in h(P_1)$. $\square$

2.4.2 Boolean algebras and linear orders

To end this section, we discuss how for computable Boolean algebras, it is impossible to create an example which witnesses the conclusion of Theorem 2.4.1 (or even the main result of [9]). Moreover, it is impossible to produce an example of a computable Boolean algebra $\mathcal{B}$ and a c.e. degree $\mathbf{d}$ such that $\mathcal{B}$ is not computably categorical but is computably categorical relative to $\mathbf{d}$. We hope to eventually show that the same holds for the class of computable linear orders.

For computable Boolean algebras, we have the following results.

Theorem 2.4.11 (Gončarov [15]). A computable Boolean algebra is computably categorical if and only if it has finitely many atoms.

In fact, this fully characterizes relative computable categoricity for computable Boolean algebras. Additionally, Bazhenov showed the following.

Theorem 2.4.12 (Bazhenov [4]). For every degree $\mathbf{d} < \mathbf{0}'$, a computable Boolean algebra is $\mathbf{d}$-computably categorical if and only if it is computably categorical.

For a computable Boolean algebra $\mathcal{B}$, if $\mathcal{B}$ is computably categorical, then it is also relatively computably categorical and so must be computably categorical relative to all $\mathbf{d} \geq \mathbf{0}$. That is, no computable computably categorical Boolean algebra can witness the nonmonotonic behavior of computably categoricity relative to a degree in either the main result of [9] or in Theorem 2.4.1.

Bazhenov's result is needed for the following variation. The structure need not be computably categorical as seen in Theorem 2.1.7, where we begin with a graph which is not computably categorical but changes to being computably categorical relative to a degree $\mathbf{d} > \mathbf{0}$. However, this behavior also cannot be witnessed by a computable Boolean algebra. Let

$\mathcal{B}$ be a computable Boolean algebra which is not computably categorical, then by Bazhenov's theorem, we have that for all degrees $\mathbf{d} < \mathbf{0}'$, $\mathcal{B}$ is *not* $\mathbf{d}$-computably categorical. That is, for each $\mathbf{d} < \mathbf{0}'$, there exists a computable copy $\mathcal{A}^{\mathbf{d}}$ of $\mathcal{B}$ such that $\mathcal{B}$ and $\mathcal{A}^{\mathbf{d}}$ are not $\mathbf{d}$-computably isomorphic. Such a computable copy is also a $\mathbf{d}$-computable copy of $\mathcal{B}$, and so $\mathcal{B}$ cannot be computably categorical relative to any $\mathbf{d} < \mathbf{0}'$.

It is unclear if the same outcome could be observed for computable linear orders which are not computably categorical, since Theorem 2.4.12 was a nontrivial result which followed from Bazhenov's methods to prove the main result in [4].

Theorem 2.4.13 (Bazhenov [4])**.** For $\mathcal{B}$ a computable Boolean algebra, $\mathcal{B}$ is Δ^0_2-categorical if and only if it is relatively Δ^0_2-categorical.

If we had a similar result for linear orders, then certainly that would imply that a computable linear order cannot be constructed to be not computably categorical but computably categorical relative to some c.e. degree $\mathbf{d} > \mathbf{0}$. We end this chapter with a question in this direction.

Question 2.4.14. Does there exist a degree $\mathbf{d} < \mathbf{0}'$ and a computable linear order L which is $\mathbf{d}$-computably categorical but is **not** computably categorical?

Chapter 3

The reverse mathematics of a topological theorem

3.1 Introduction

This chapter contains my contributions from joint work [5] with my advisers Reed Solomon and Damir Dzhafarov, and fellow PhD students Heidi Benham and Andrew DeLapo.

3.1.1 Brief overview of reverse mathematics

Reverse mathematics is a research program in mathematical logic that aims to study which logical axioms are both sufficient and necessary to prove mathematical theorems. The logical strength of a mathematical theorem is measured using subsystems of second-order arithmetic. In particular, the base theory over which we prove reversals is called RCA_0.

Definition 3.1.1. The formal system RCA_0 consists of the following axioms and axiom schema:

(1) PA^-, i.e., axioms which describe a discrete ordered semiring;

(2) the Δ^0_1 comprehension scheme; and

(3) $\mathsf{I}\Sigma^0_1$ (the induction axiom restricted to Σ^0_1 formulas),

where the Δ^0_1 comprehension scheme consists of the universal closure of each axiom of the form

$$(\forall n)[\varphi(n) \leftrightarrow \psi(n)] \rightarrow (\exists X)[n \in X \leftrightarrow \varphi(n)],$$

where φ is Σ^0_1 and ψ is Π^0_1.

In RCA_0, we can formalize many of the coding methods utilized in computable mathematics, and so we can show some mathematical theorems are provable in RCA_0.

Theorem 3.1.2 ([29])**.** The following mathematical theorems are provable in RCA_0:

(1) The Baire category theorem;

(2) the existence of an algebraic closure of a countable field;

(3) and the intermediate value theorem.

However, not every theorem is provable in RCA_0, and in order to measure their logical strength, we consider stronger subsystems such as ACA_0, which we obtain by allowing the comprehension scheme to be used for all arithmetical formulas.

Definition 3.1.3. The formal system ACA_0 consists of RCA_0 and the comprehension scheme for all arithmetical formulas.

Over RCA_0, we can show that several mathematical theorems are equivalent to ACA_0 by using the following fact.

Theorem 3.1.4 (RCA_0)**.** The following are equivalent:

(1) ACA_0.

(2) For every 1-to-1 function $f : \mathbb{N} \to \mathbb{N}$, the range of f exists, i.e., the set

$$\mathrm{rg}(f) = \{m : \exists n \in \mathbb{N}(f(n) = m)\}$$

exists.

Many well-known mathematical theorems are equivalent, over RCA_0, to one of the following subsystems, listed in order of increasing provability strength: WKL_0, ACA_0, ATR_0, Π^1_1-CA_0. However, recent developments in reverse mathematics has shown that combinatorial results form a rich zoo beneath ACA_0, beginning with results which show that the principle RT^2_2 is strictly weaker than ACA_0 [27] and does not imply WKL_0 [23].

Definition 3.1.5. Let $[\mathbb{N}]^n$ denote the collection of n-element subsets of $\mathbb{N}$. A k**-coloring** of $[\mathbb{N}]^n$ is a map $c : [\mathbb{N}]^n \to k$. A set $H \subseteq \mathbb{N}$ is **homogeneous** for c if there is an $i < k$ where $c(s) = i$ for all $s \in [H]^n$.

Definition 3.1.6. RT^2_2 is the statement that every 2-coloring $c : [\mathbb{N}]^2 \to 2$ admits an infinite homogeneous set H.

For the results mentioned in this chapter, we define two specific combinatorial principles, both being consequences of RT^2_2.

Definition 3.1.7 (Chain/antichain principle)**.** CAC is the statement that every infinite partial order $(P, \leq_P)$ has an infinite chain or antichain.

Definition 3.1.8 (Ascending/descending sequence principle)**.** ADS is the statement that every infinite linear order has an infinite ascending sequence or an infinite descending sequence.

We have that over RCA_0, RT^2_2 strictly implies CAC [20] and CAC strictly implies ADS [22]. For a more general background on reverse mathematics, see Simpson [29] or Dzhafarov and Mummert [10].

3.1.2 Preliminaries from topology

In this section, we recall some definitions from topology that will appear frequently throughout this chapter. We first begin with the separation axioms.

Definition 3.1.9. Let X be a topological space.

(1) X is said to be T_0 if for any distinct $x, y \in X$, there is an open set U such that $x \in U$ and $y \notin U$, or $x \notin U$ and $y \in U$.

(2) X is said to be T_1 if for any distinct $x, y \in X$, there are open sets U and V such that $x \in U$ and $y \notin U$, and $x \notin V$ and $y \in V$.

(3) X is said to be T_2 if for any distinct $x, y \in X$, there are open sets U and V such that $x \in U$ and $y \notin U$, $x \notin V$ and $y \in V$, and $U \cap V = \emptyset$.

Note that a T_2 space is more commonly referred to as a Hausdorff space. We also note that if a space is T_2, then it is also T_1 and T_0. There are various examples of topological spaces that satisfy weaker separation axioms but not the stronger separation axioms. One such example is the Sierpiński space, whose underlying set is $\{0, 1\}$ and the open sets are $\emptyset$, $\{1\}$, and $\{0, 1\}$. This space is T_0 but not T_1.

In order to study topological spaces in the context of reverse math, we will use a formalization of countable spaces (see section 3.1.3) where we can specify the open sets in the topology via a countable basis.

Definition 3.1.10. X is said to be **second-countable** (or is a **second-countable space**) if there is a countable collection $\mathcal{U} = \{U_i\}_{i \in \mathbb{N}}$ of open subsets of X that form a basis for the topology on X.

It is not true in general that every countable space will also be second-countable, but we will narrow our focus on countable spaces which are second-countable in order to use the codings available in RCA_0.

3.1.3 Brief overview of CSC spaces

Ginsburg and Sands proved the following topological theorem in [14] using a combinatorial proof with applications of the principles CAC and ADS.

Theorem 3.1.11 (Ginsburg-Sands)**.** Every infinite topological space contains one of the following five spaces, with $\mathbb{N}$ as the underlying set, as a subspace:

(i) *discrete*: all subsets of $\mathbb{N}$ are open;

(ii) *indiscrete*: the only open sets are $\mathbb{N}$ and $\emptyset$;

(iii) *cofinite*: the open sets are $\mathbb{N}$, $\emptyset$, and all subsets of $\mathbb{N}$ with finite complement;

(iv) *initial segment*: the open sets are $\mathbb{N}$, $\emptyset$, and all sets of the form $[0, n] = \{k \in \mathbb{N} : k \leq n\}$;

(v) *final segment*: the open sets are $\mathbb{N}$, $\emptyset$, and all sets of the form $[n, \infty) = \{k \in \mathbb{N} : n \leq k\}$.

Moreover, no two of the five spaces above are homeomorphic, and each of the spaces is homeomorphic to all of its infinite subspaces. Hence, such a space from above is said to be a **minimal space** within the original infinite space. In order to study Theorem 3.1.11 in a reverse math setting, we restricted the theorem to presentations of topological spaces known as CSC spaces, due to Dorais [7].

Definition 3.1.12. A **countable second-countable (CSC)** space is a tuple $\langle X, \mathcal{U}, k\rangle$ as follows:

(1) X is a subset of $\mathbb{N}$;

(2) $\mathcal{U} = \langle U_n : n \in \mathbb{N}\rangle$ is a family of subsets of X such that every $x \in X$ belongs to U_n for some $n \in \mathbb{N}$;

(3) $k : X \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is a function such that for every $x \in X$ and all $m, n \in \mathbb{N}$, if $x \in U_m \cap U_n$ then $x \in U_{k(x,m,n)} \subseteq U_n \cap U_m$.

Here, $\mathcal{U}$ is the **basis** for $\langle X, \mathcal{U}, k\rangle$ if it is closed under finite intersections. If $\mathcal{U}$ is not closed under finite intersections, then it is only a subbasis, but which can be extended to a basis. On this extension, we can define our k function to satisfy (3) by Lemma 3.2 in [5].

We say that $\langle X, \mathcal{U}, k\rangle$ is an **infinite** CSC space if X is infinite. Although the function k is part of the presentation of a CSC space, we can essentially build CSC spaces in RCA_0 by specifying a countable basis of basic open sets U_n, where these sets can originate from an arbitrary collection of sets.

Proposition 3.1.13 (Proposition 2.12, [7])**.** The following is provable in RCA_0. Given a set $X \subseteq \mathbb{N}$ and a collection $\langle V_n : n \in \mathbb{N}\rangle$ of subsets of X, there exists a CSC space $\langle X, \mathcal{U}, k\rangle$ with $\mathcal{U} = \{U_n : n \in \mathbb{N}\}$ as follows:

(1) for every $n \in \mathbb{N}$, $V_n \in \mathcal{U}$;

(2) for every $m \in \mathbb{N}$, $U_m = \bigcap_{n \in F} V_n$, where F is the finite set coded by m.

We call $\langle X, \mathcal{U}, k\rangle$ above the **CSC space generated by** $\langle V_n : n \in \mathbb{N}\rangle$. Additionally, since every finite set is coded by a number, sets in $\mathcal{U}$ are closed under finite intersections, and so we can obtain our function k for the CSC space. Using this fact, we can build topological spaces with certain properties by specifying $\langle V_n : n \in \mathbb{N}\rangle$. Throughout this chapter, we will use "$X \in \mathcal{U}$" as shorthand for the formula $\exists n(X = U_n \wedge U_n \in \mathcal{U})$.

We now give some classical definitions from topology in the specific context of CSC spaces.

Definition 3.1.14. Let $\langle X, \mathcal{U}, k\rangle$ be a CSC space with $\mathcal{U} = \{U_n : n \in \mathbb{N}\}$.

(1) For $Y \subseteq X$, let $\mathcal{U} \upharpoonright Y = \langle U_n \cap Y : n \in \mathbb{N}\rangle$.

(2) A **subspace** of $\langle X, \mathcal{U}, k\rangle$ is a tuple $\langle Y, \mathcal{U} \upharpoonright Y, k\rangle$ for some $Y \subseteq X$.

Definition 3.1.15. Let $\langle X, \mathcal{U}, k\rangle$ be a CSC space.

(1) X is T_0 if for $x \neq y$ in X, there exists $U \in \mathcal{U}$ such that either $x \in U$ and $y \notin U$, or $x \notin U$ and $y \in U$.

(2) X is T_1 is for $x \neq y$ in X, there exists $U, V \in \mathcal{U}$ such that $x \in U \setminus V$ and $y \in V \setminus U$.

In RCA_0, it is fairly straightforward to prove that given a CSC space X and $Y \subseteq X$, the subspace $\langle Y, \mathcal{U} \upharpoonright Y, k \rangle$ exists. It is also easy to prove in RCA_0 that a given T_1 CSC space X must also be T_0. We now give definitions for specific topologies on our CSC spaces.

Definition 3.1.16. Let $\langle X, \mathcal{U}, k \rangle$ be a CSC space. X is said to be **indiscrete** if $U \in \mathcal{U}$ if and only if $U = \emptyset$ or $U = X$.

Definition 3.1.17. Let $\langle X, \mathcal{U}, k \rangle$ be a CSC space.

(1) X has the **initial segment** topology if there is a bijection $h : \mathbb{N} \to X$ such that $U \in \mathcal{U}$ if and only if U is $\emptyset$, X, or $\{h(i) : i \leq j\}$ for some $j \in \mathbb{N}$.

(2) X has the **final segment** topology if there is a bijection $h : \mathbb{N} \to X$ such that $U \in \mathcal{U}$ if and only if U is $\emptyset$, X, or $\{h(i) : i \geq j\}$ for some $j \in \mathbb{N}$.

We will refer to the bijection h above as a **homeomorphism** like in classical topology. Since we are stating the existence of a homeomorphism h, these definitions may seem stronger than they need to be. There are other ways of defining these topologies in terms of just open sets without needing to mention a map h.

Definition 3.1.18. Let $\langle X, \mathcal{U}, k \rangle$ be an infinite CSC space.

(1) X has the **weak initial segment** topology if:

  (a) every $U \in \mathcal{U}$ is finite or equal to X;

  (b) if $U, V \in \mathcal{U}$ then either $U \subseteq V$ or $V \subseteq U$;

  (c) for each $s \in \mathbb{N}$, there is a finite $U \in \mathcal{U}$ such that $|U| = s$;

  (d) each $x \in X$ belongs to some finite $U \in \mathcal{U}$.

(2) X has the **weak final segment** topology if:

  (a) every $U \in \mathcal{U}$ is cofinite in X or is $\emptyset$;

  (b) if $U, V \in \mathcal{U}$ then either $U \subseteq V$ or $V \subseteq U$;

(c) for each $s \in \mathbb{N}$, there is a nonempty $U \in \mathcal{U}$ such that $|X \setminus U| = s$;

(d) each $x \in X$ belongs to some $U \neq X \in \mathcal{U}$.

However, in RCA_0, it turns out that we can use the stronger Definition 3.1.17 without inflating a principle's strength for the purposes of the results in this chapter. Recall that for the Ginsburg-Sands theorem, we are interested in finding infinite subspaces with certain topologies.

Proposition 3.1.19 (Proposition 3.14, [5])**.** The following is provable in RCA_0.

(1) Every infinite CSC space $\langle X, \mathcal{U}, k\rangle$ with the weak initial segment topology has an infinite subspace with the initial segment topology.

(2) Every infinite CSC space $\langle X, \mathcal{U}, k\rangle$ with the weak final segment topology has an infinite subspace with the final segment topology.

Hence, we can begin with a CSC space $\langle X, \mathcal{U}, k\rangle$ with the weak initial or final segment topology, but when we pass to the infinite subspace with the corresponding stronger topology, we can utilize homeomorphisms in our proofs when needed.

3.2 Without and with the closure relation

3.2.1 The existence of the closure relation and ACA_0

Here, we sketch only a part of the classical argument for Ginsburg-Sands. Let X be an infinite topological space, then we can define the following equivalence relation on X:

$$x \sim y \iff \mathrm{cl}\{x\} = \mathrm{cl}\{y\}.$$

If there is an infinite equivalence class, then we are done as we have obtained an infinite subspace with the indiscrete topology. Otherwise, each equivalence class is finite, and so there must be infinitely many classes. By choosing one point from each class, we form an infinite subspace which is T_0. This subspace is T_0 because if $x \neq y$, then $\mathrm{cl}\{x\} \neq \mathrm{cl}\{y\}$ and so $\mathrm{cl}\{x\}$ is a closed set containing x but not y or $\mathrm{cl}\{y\}$ is a closed set containing y but not x. On this T_0 subspace, we define the following partial order:

$$x \leq y \iff x \in \mathrm{cl}\{y\}.$$

With this $\leq$-ordering, we obtain an infinite partially-ordered set where we can apply CAC to obtain either an infinite antichain or an infinite chain. If we have an infinite chain, we can apply ADS to obtain either an infinite ascending sequence or an infinite descending sequence. In the former, we obtain an infinite subspace satisfying (v), and in the latter, we obtain an infinite subspace satisfying (iv). If instead we have an infinite antichain, then this forms an infinite T_1 subspace of X. This case requires a separate argument to prove Theorem 3.1.11 which does not involve CAC or ADS (see [5, Section 2]).

To begin studying the logical strength of Theorem 3.1.11, we begin with the following classical definition.

Definition 3.2.1. The **closure relation** cl_X on a topological space X is the binary relation defined by

$$(y, x) \in \mathrm{cl}_X \iff y \in \mathrm{cl}\{x\}.$$

We now formalize the closure relation on a CSC space in RCA_0.

Definition 3.2.2 (RCA_0)**.** The **closure relation** cl_X on a CSC space X is the binary relation defined by

$$(y, x) \in \mathrm{cl}_X \iff (\forall n)(y \in U_n \to x \in U_n).$$

Classically, we can define the closure of a point $x \in X$ as the set

$$\mathrm{cl}\{x\} = \{y \in X : \text{for every basic open set } U,\ (y \in U \to x \in U)\}.$$

Since we are given the basic open sets U_n for a CSC space X, Definition 3.2.2 fully captures the closure of a point in X. We now show the exact logical strength needed to prove that

cl_X exists for any given CSC space X.

Theorem 3.2.3 (RCA_0)**.** The following are equivalent:

(1) ACA_0.

(2) For a CSC space $\langle X, \mathcal{U}, k\rangle$, the closure relation cl_X exists.

Proof. (1) implies (2) is immediate by Definition 3.2.2. For the converse, we will use Theorem 3.1.4. Let $f : \mathbb{N} \to \mathbb{N}$ be an injective function. For each $n = \langle e, s\rangle$, we define the following sets:

$$U_n = \begin{cases} \{2e, 2e+1\} & \text{if } (\forall m \leq s)(f(m) \neq e) \\ \{2e\} & \text{if } (\exists m \leq s)(f(m) = e). \end{cases}$$

We claim that $e \in \mathrm{rg}(f)$ if and only if $2e \notin \mathrm{cl}_X(2e+1)$. If $2e \notin \mathrm{cl}_X(2e+1)$, then there exists a basic open set U_n such that $2e \in U_n$ but $2e+1 \notin U_n$. In particular, if $n = \langle e, s\rangle$, we have that $U_{\langle e,s\rangle} = \{2e\}$, and so there exists an $m \leq s$ such that $f(m) = e$. If $2e \in \mathrm{cl}_X(2e+1)$, we have that for all basic open sets U_n where $2e \in U_n$, it must also be the case that $2e+1 \in U_n$. So, for all s where $n = \langle e, s\rangle$, we have that $U_n = \{2e, 2e+1\}$. Hence, for all s, we have that for all $m \leq s$ that $f(m) \neq e$, and thus $e \notin \mathrm{rg}(f)$. $\square$

By Theorem 3.2.3, we have that a proof of the Ginsburg-Sands theorem which uses the closure relation must assume at least ACA_0. We now turn our attention to the case where the closure operation is given as part of the representation for a CSC space.

3.2.2 Weak Ginsburg-Sands with the closure relation ($\mathsf{wGS}^{\mathrm{cl}}$) and CAC

Consider the following new principle.

$\mathsf{wGS}^{\mathrm{cl}}$: Let $\langle X, \mathcal{U}, k\rangle$ be an infinite CSC space with a closure relation cl_X. Then, X has one of the following:

(i) an infinite T_1 subspace;

(ii) an infinite indiscrete subspace;

(iii) an infinite subspace homeomorphic to $\mathbb{N}$ with the initial segment topology;

(iv) an infinite subspace homeomorphic to $\mathbb{N}$ with the final segment topology.

$\mathsf{wGS}^{\mathrm{cl}}$ is a "weakening" of the original Ginsburg-Sands theorem in the sense that the infinite discrete subspace or infinite subspace with the cofinite topology cases are collapsed into the singular T_1 case in (i). The following result shows that we have a topological characterization of the combinatorial principle CAC.

Theorem 3.2.4 (RCA_0)**.** The following are equivalent:

(1) CAC.

(2) $\mathsf{wGS}^{\mathrm{cl}}$.

Proof. For (1) implies (2), let X be a CSC space with its closure relation cl_X given. We follow the same line of argument as in section 3.2 to obtain a T_0 subspace of X where we can define a partial order by saying that

$$x \leq y \iff x \in \mathrm{cl}\{y\}.$$

Our T_0 subspace, call it Y, with this partial ordering is an infinite partially-ordered set, and so we can use CAC to obtain either an infinite antichain or an infinite chain. If we have an infinite antichain, this forms an infinite T_1 subspace of X since for each $x, y \in Y$, $x \not\leq y$ and $y \not\leq x$, and so we have open sets in X which separate the two points from each other by the definition of $\leq$.

If we have an infinite chain, then we can apply ADS to obtain either an infinite ascending sequence or an infinite descending sequence. Let $y_0 < y_1 < y_2 < \ldots$ be an infinite ascending sequence in Y. Note that this is strictly increasing because Y is T_0. So for each $i \in \mathbb{N}$, there is an open subset of X which contains y_{i+1} but not y_i, but every open set containing y_i also contains y_{i+1}. Let A be a nonempty open subset of Y and we define $m = \min\{i \in \mathbb{N} : y_i \in A\}$.

We have that $A = \{y_i : i \geq m\}$ by the argument in the previous sentence, and so each set of the form $\{y_i : i \geq n\}$ is open in Y for each n. Hence, Y with the subspace topology is homeomorphic to $\mathbb{N}$ with the final segment topology via the homeomorphism $h : Y \to \mathbb{N}$ defined by $h(y_i) = i$. If we started with an infinite descending sequence, a symmetric argument yields us an infinite subspace homeomorphic to $\mathbb{N}$ with the initial segment topology.

We now prove that (2) implies (1) over RCA_0. Let $P = (\omega, \leq_P)$ be an infinite partially-ordered set. We form a CSC space $X = \langle P, \mathcal{U}, k\rangle$ generated by $\{U_p\}_{p \in P}$ where

$$U_p = \{x \in P : p \leq_P x\}$$

For this order topology, we have that for each $x \in P$ that $\mathrm{cl}\{x\} = \{p \in P : p \leq_P x\}$. Additionally, we have that for any open set U in X, if $p \in U$ and for $x \in P$ if we have that $p \leq_P x$, then $x \in U$ as well. That is, open sets in X are upwards closed relative to the $\leq_P$-ordering.

If X has an infinite T_1 subspace H, then we have for all $h \in H$, $\mathrm{cl}\{h\} \cap H = \{h\}$. So, for any $x \in H$ such that $x \neq h$, we have that $x \ngeq_P h$ and $h \ngeq_P x$. So, H forms an infinite antichain.

We now prove the following lemma.

Lemma 3.2.5. *X cannot have an infinite indiscrete subspace.*

Proof. Suppose H is an infinite indiscrete subspace of X. Let $x \neq y$ be elements of H and without loss of generality, assume that $x \ngeq_P y$, and so either $x <_P y$ or x and y are incomparable. In either case, let U be the basic open set $\{p \in P : y \leq_p p\}$. Since $y \in U$ and $x \notin U$, we have that $U \cap H$ is a proper nonempty open set in the subspace topology. □

By this lemma, we can rule out (ii) from $\mathsf{wGS}^{\mathrm{cl}}$, as it will never occur for our particular space X.

Finally, suppose X has an infinite subspace homeomorphic to $\mathbb{N}$ with the final segment topology via a homeomorphism $\varphi : \mathbb{N} \to H$ where $\varphi(i) = h_i$ for $i \in \mathbb{N}$. Consider the two points h_i and h_{i+1} from H, then we have that

$$\varphi([i+1,\infty)) = H \cap V_{i+1}$$

where V_{i+1} is an open set in X. If $h_{i+1} <_P h_i$, then because $h_{i+1} \in V_{i+1}$, it follows that $h_i \in V_{i+1}$. But, $h_i \notin \varphi([i+1,\infty))$ because $i <_{\mathbb{N}} i+1$ where $<_{\mathbb{N}}$ is the standard ordering on $\mathbb{N}$, so $h_i \notin V_{i+1}$.

If instead we have that h_i and h_{i+1} are incomparable in P, then there is an open set V_i in X where $h_i \in V_i$ but $h_{i+1} \notin V_i$ and an open set V_{i+1} in X where $h_{i+1} \in V_{i+1}$ but $h_i \notin V_{i+1}$. By φ, we can write $V_i \cap H = \varphi([k_0,\infty))$ for some $k_0 \leq_{\mathbb{N}} i$ and $V_{i+1} \cap H = \varphi([k_1,\infty))$ for some $k_1 \leq_{\mathbb{N}} i+1$. But $k_0 \leq_{\mathbb{N}} i \leq_{\mathbb{N}} i+1$, and so $h_{i+1} \in V_i \cap H$, but V_i was an open set which did not contain h_{i+1}.

Hence, it must be the case that $h_i <_P h_{i+1}$ and so we can form an infinite ascending chain with respect to $\leq_P$ in H. The case where X has an infinite subspace homeomorphic to $\mathbb{N}$ with the initial segment topology is symmetric and will yield an infinite descending chain in P. $\square$

Bibliography

[1] B. Anderson and B. Csima. “Degrees That Are Not Degrees of Categoricity”. *Notre Dame Journal of Formal Logic* 57 (2016), 389–398.

[2] C. J. Ash. *Computable Structures and the Hyperarithmetical Hierarchy.* Ed. by J. Knight. New York: Elsevier, 2000.

[3] C. Ash, J. Knight, M. Manasse, and T. Slaman. “Generic copies of countable structures”. *Annals of Pure and Applied Logic* 42 (1989), 195–205.

[4] N. A. Bazhenov. “Δ^0_2-Categoricity of Boolean Algebras”. *Journal of Mathematical Sciences* 203 (2014), 444–454.

[5] H. Benham, A. DeLapo, D. D. Dzhafarov, R. Solomon, and J. D. Villano. “The Ginsburg–Sands theorem and computability theory”. *Advances in Mathematics* 444 (2024), 109618.

[6] J. Chisholm. “Effective Model Theory vs. Recursive Model Theory”. *The Journal of Symbolic Logic* 55 (1990), 1168–1191.

[7] F. G. Dorais. *Reverse mathematics of compact countable second-countable spaces.* 2011.

[8] R. Downey. “On Presentations of Algebraic Structures”. *Complexity, Logic, and Recursion Theory* 187 (June 2003).

[9] R. Downey, M. Harrison-Trainor, and A. Melnikov. “Relativizing computable categoricity”. *Proc. Amer. Math. Soc.* 149 (2021), 3999–4013.

[10] D. Dzhafarov and C. Mummert. *Reverse Mathematics: Problems, Reductions, and Proofs.* Theory and Applications of Computability. Springer International Publishing, 2022.

[11] J. L. Eršov. “Theorie Der Numerierungen III”. *Mathematical Logic Quarterly* 23 (1977), 289–371.

[12] J. N. Y. Franklin. “Strength and Weakness in Computable Structure Theory”. *Computability and Complexity: Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday.* Ed. by A. Day, M. Fellows, N. Greenberg, B. Khoussainov, A. Melnikov, and F. Rosamond. Cham: Springer International Publishing, 2017, 302–323.

[13] J. N. Y. Franklin and R. Solomon. “Degrees that Are Low for Isomorphism”. *Computability* 3 (2014), 73–89.

[14] J. Ginsburg and B. Sands. "Minimal Infinite Topological Spaces". *The American Mathematical Monthly* 86 (1979), 574–576.

[15] S. S. Gončarov. "Some properties of the constructivization of Boolean algebras". *Siberian Mathematical Journal* 16 (1975), 203–214.

[16] S. S. Gončarov. "The problem of the number of nonautoequivalent constructivizations". *Algebra i Logika* 19 (1980), 621–639, 745.

[17] S. S. Gončarov. "The quantity of nonautoequivalent constructivizations". *Algebra and Logic* 16 (May 1977), 169–185.

[18] S. S. Gončarov, S. Lempp, and R. Solomon. "The computable dimension of ordered abelian groups". *Advances in Mathematics* 175 (2003), 102–143.

[19] D. R. Hirschfeldt, B. Khoussainov, R. A. Shore, and A. M. Slinko. "Degree spectra and computable dimensions in algebraic structures". *Annals of Pure and Applied Logic* 115 (2002), 71–113.

[20] D. R. Hirschfeldt and R. A. Shore. "Combinatorial principles weaker than Ramsey's Theorem for pairs". *Journal of Symbolic Logic* 72 (2007), 171–206.

[21] O. V. Kudinov. "An autostable 1-decidable model without a computable Scott family of $\exists$-formulas". *Algebra and Logic* 35 (July 1996), 255–260.

[22] M. Lerman, R. Solomon, and H. Towsner. "Separating principles below Ramsey's Theorem for Pairs". *Journal of Mathematical Logic* 13 (Feb. 2013).

[23] J. Liu. "RT^2_2 does not imply WKL_0". *The Journal of Symbolic Logic* 77 (2012), 609–620.

[24] C. McCoy. "Partial Results in Δ^0_3-Categoricity in Linear Orderings and Boolean Algebras". *Algebra and Logic* 41 (Sept. 2002), 295–305.

[25] C. F. D. McCoy. "Δ^0_2-Categoricity in Boolean Algebras and Linear Orderings". *Annals of Pure and Applied Logic* 119 (2003), 85–120.

[26] J. B. Remmel. "Recursively Categorical Linear Orderings". *Proc. Amer. Math. Soc.* 83 (1981), 387–391.

[27] D. Seetapun and T. A. Slaman. "On the Strength of Ramsey's Theorem". *Notre Dame Journal of Formal Logic* 36 (1995), 570–582.

[28] V. L. Selivanov. "Enumerations of families of general recursive functions". *Algebra and Logic* 15 (Mar. 1976), 128–141.

[29] S. G. Simpson. *Subsystems of Second Order Arithmetic.* 2nd ed. Perspectives in Logic. Cambridge University Press, 2009.

[30] J. D. Villano. *Computable categoricity relative to a c.e. degree.* 2024.