THE $\Delta^0_2$ TURING DEGREES: AUTOMORPHISMS AND DEFINABILITY

THEODORE A. SLAMAN AND MARIYA I. SOSKOVA

Abstract. We prove that the $\Delta^0_2$ Turing degrees have a finite automorphism base. We apply this result to show that the automorphism group of $\mathcal{D}_T(\leq \bar{0}')$ is countable and that all its members have arithmetic presentations. We prove that every relation on $\mathcal{D}_T(\leq \bar{0}')$ induced by an arithmetically definable degree invariant relation is definable with finitely many $\Delta^0_2$ parameters and show that rigidity for $\mathcal{D}_T(\leq \bar{0}')$ is equivalent to its biinterpretability with first order arithmetic.

1. Introduction

We wish to understand the notion of relative definability between sets of natural numbers and in particular its natural presentation as a degree structure. There are many ways in which one might approach this task. Here we consider the following three questions as guiding: What is the expressive power of the theory of the degree structure? What are the definable relations in the structure? How can we describe its automorphism group? For the structure of the Turing degrees $\mathcal{D}_T$ we have a complete answer to the first question and an appealing conjecture for the other two. Simpson [6] proved that the first order theory of the Turing degrees is computably isomorphic to the theory of second order arithmetic. In [7], Slaman and Woodin conjecture that the relationship between the structure of the Turing degrees and second order arithmetic is much stronger. If true their biinterpretability conjecture gives complete answers to the second and third questions: the definable relations in $\mathcal{D}_T$ are exactly the ones induced by degree invariant relations definable in second order arithmetic and the only automorphism of $\mathcal{D}_T$ is the identity. In other words the biinterpretability conjecture is that the structure of the Turing degrees is logically equivalent to second order arithmetic. Slaman and Woodin come to this conjecture through their work on the definability and automorphisms of the Turing degrees described in [7]. They establish that the structure of the Turing degrees has a finite automorphism base and as a consequence obtain that the biinterpretability conjecture is true if we allow the use of finitely many parameters, that the automorphism group of the Turing degrees is countable and that every member has an arithmetically definable presentation. Furthermore they show that every relation in $\mathcal{D}_T$ induced by degree invariant relations definable in second order arithmetic is first order definable with parameters in $\mathcal{D}_T$. Finally they establish that the rigidity of $\mathcal{D}_T$, the statement that $\mathcal{D}_T$ has no nontrivial automorphisms, is equivalent to its biinterpretability with second order arithmetic.

The first author was partially supported by National Science Foundation grant number DMS-1301659. The second author was partially supported by an FP7-MC-IOF grant STRIDE (298471) and by the Sofia university Science fund.
In parallel to the study of the global structure of the Turing degrees, two substructures are investigated: the structure of the Turing degrees bounded by the degree of the halting problem $\mathbf{0}'$ and the substructure of the computably enumerable degrees, denoted respectively by $\mathcal{D}_{PT}(\leq \mathbf{0}')$ and $\mathcal{R}$. The same three guiding questions can be addressed to understand these local structures. In this case as well we have complete answers to the first one. Both structures have theories that are computably isomorphic to first order arithmetic. Shore [4] proves this for $\mathcal{D}_{PT}(\leq \mathbf{0}')$ and Slaman and Harrington (see [2]) prove it for $\mathcal{R}$. The second question is understood only to a certain extent. Nies, Shore and Slaman [2] prove that any relation in $\mathcal{R}$ that is induced by a definable relation in first order arithmetic, invariant not only under Turing reducibility, but also under double jump, is definable in $\mathcal{R}$. Later Shore [5] proves that this holds also for $\mathcal{D}_{PT}(\leq \mathbf{0}')$. Even less is known regarding the third question.

In this article we focus on the structure $\mathcal{D}_{PT}(\leq \mathbf{0}')$. We establish a relationship between the local structure $\mathcal{D}_{PT}(\leq \mathbf{0}')$ and first order arithmetic, similar to the one proved by Slaman and Woodin for the global structure $\mathcal{D}_{PT}$ and second order arithmetic. Our main result is that $\mathcal{D}_{PT}(\leq \mathbf{0}')$ has a finite automorphism base and it is biinterpretable with first order arithmetic using the elements of this base as parameters. We further show that $\text{Aut}(\mathcal{D}_{PT}(\leq \mathbf{0}'))$ is countable and its members have arithmetic presentations and that the structure $\mathcal{D}_{PT}(\leq \mathbf{0}')$ is atomic. We answer Question 4.8 from [9]: every relation in $\mathcal{D}_{PT}(\leq \mathbf{0}')$ induced by an arithmetically definable degree invariant relation is definable with finitely many parameters. Finally, we show that rigidity for $\mathcal{D}_{PT}(\leq \mathbf{0}')$ is equivalent to its biinterpretability with first order arithmetic.

In a sequel to this paper, [8], we use similar methods to investigate the structure of the enumeration degrees. There we reveal a strong connection between the local and global structures: the natural presentation of the structure of the computably enumerable degrees is an automorphism base for the structure of the enumeration degrees.

2. The local coding theorem and its applications

One of the main tools in Slaman and Woodin’s analysis of the automorphisms of $\mathcal{D}_{PT}$ is their coding theorem, the statement that every countable relation in the Turing degrees is uniformly definable from parameters. For the local structure $\mathcal{D}_{PT}(\leq \mathbf{0}')$ Slaman and Woodin give a more restricted version of the coding theorem. In order to state it we need the following definitions related to lowness. We use the notation $\{i\}^A$ to denote the $i$-th partial computable function with oracle $A$ in some standard effective listing of all partial computable functions.

**Definition 1.**

1. A set $A$ is low if its jump $A'$ is computable from the halting set. A Turing degree is low if it contains a low set.
2. A sequence of sets $\{Z_i\}_{i<\omega}$ is uniformly computable from a set $A$ if there is a computable function $f$, such that $Z_i = \{ f(i) \}^A$.
3. A sequence of sets $\{Z_i\}_{i<\omega}$ is uniformly low if it is uniformly computable from the halting set $\emptyset'$ and there is a computable function $g$, such that $(\bigoplus_{i<\omega} Z_i)' = \{ g(i) \}^{\emptyset'}$.
4. A set of degrees $\mathcal{Z} \subseteq \mathcal{D}_{PT}(\leq \mathbf{0}')$ is uniformly low if there is a uniformly low sequence $\{Z_i\}_{i<\omega}$, such that $\mathcal{Z} = \{ d_T(Z_i) \mid i < \omega \}$. We call the sequence $\{Z_i\}_{i<\omega}$ a presentation of $\mathcal{Z}$. 
We give an example of a uniformly low set of degrees. Let \( \{ G_i \}_{i < \omega} \) be a sequence of sets such that \( G = \bigoplus_{i < \omega} G_i \) is low. Then the set \( \mathcal{G} = \{ d_T(G_i) \mid i < \omega \} \) is uniformly low. In particular if \( G \) is any 1-generic set then \( \mathcal{G} \) is a uniformly low anti-chain. We refer to [7] for a review of properties of 1-generic sets, some of which will be used in this article.

The local coding theorem of Slaman and Woodin [9] is that every uniformly low set that is bounded by a low degree is uniformly definable from parameters in \( D_T(\leq 0') \). Among many other applications, it gives a method for coding a model of arithmetic in \( D_T(\leq 0') \). We can represent a model of arithmetic as a partial ordering as described in [2]. We start with a countable anti-chain of minimal elements \( \{ p_n \}_{n < \omega} \) which will represent the natural numbers. Then, for each \( n, m \in \omega \) we add an element \( c_{n,m} \) which represents the pair \((p_n, p_m)\). Next we add ascending chains of lengths 2 and 3, respectively, from \( p_n \) to \( c_{n,m} \) and from \( p_m \) to \( c_{n,m} \). Finally, to code addition, we add a chain of length 4 from \( p_{n+m} \) to \( c_{n,m} \) and for multiplication, a chain of length 5 from \( p_{nm} \) to \( c_{n,m} \).

Now we fix a 1-generic set \( G \) and let \( G = \bigoplus_{i < \omega} G_i \). Recall that \( \{ G_i \}_{i < \omega} \) has the following property: if \( F \) and \( E \) are finite sets then \( \bigoplus_{i \in F} G_i \leq_T \bigoplus_{i \in E} G_i \) if and only if \( F \subseteq E \). We use this fact to build six anti-chains that together form the partial order described above. The minimal elements form one anti chain, the maximal elements another and we have four intermediate anti-chains \( L_1, L_2, L_3 \) and \( L_4 \). We partition \( \mathbb{N} \) into six computable infinite pieces \( D_i, i = 1 \ldots 6 \). To make notation easier we assume that \( D_1 = 2\mathbb{N} \). We use \( G_{2n} \) to represent \( p_n \). This will be the first anti-chain. To every pair of natural numbers \( (n, m) \) we computably assign four elements with indices in \( D_2 \), three elements with indices in \( D_3 \), two with indices in \( D_4 \) and one element with an index in each of \( D_5 \) and \( D_6 \). To construct the second anti-chain \( L_1 \) for every pair of natural numbers \( (n, m) \) we join \( G_{2n} \) with the first
assigned element from $D_2$, $G_{2m}$ with the second, $G_{2(n+m)}$ and $G_{2n+m}$ with the third and fourth. Similarly $L_2$ is constructed by joining each of the elements above $G_{2m}$, $G_{2(n+m)}$ and $G_{2n+m}$ from $L_1$ with the assigned elements from $D_3$, $L_3$ by joining the elements from $D_4$ to the $L_2$ elements above $G_{2(n+m)}$ and $G_{2n+m}$, $L_4$ by joining the element from $D_5$ to the $L_3$ element above $G_{2n+m}$. To get the final sixth anti-chain and the representatives for $c_{n,m}$ we join up all of the constructed element with the one assigned to the pair $(n,m)$ from $D_6$. Each of the six anti-chains is a uniformly low sequence bounded by $G$. We have a presentation of a partial order $P$ which by the Coding Theorem is uniformly definable from parameters $\tilde{p}$ in $D_T(\leq 0')$.

We will denote by $M(\tilde{p}) = (\mathbb{N}^{M(\tilde{p})}, 0^{M(\tilde{p})}, 1^{M(\tilde{p})}, +^{M(\tilde{p})}, \cdot^{M(\tilde{p})})$ the standard model of arithmetic coded in this way: $\mathbb{N}^{M(\tilde{p})}$ is the set of minimal elements in $P$, $0^{M(\tilde{p})}$ and $1^{M(\tilde{p})}$ are the elements representing $p_0$ and $p_1$ respectively, $M(\tilde{p}) \models n + m = k$ and $M(\tilde{p}) \models n \cdot m = l$ if and only if there is maximal element in $P$ at distance 1 from $n^{M(\tilde{p})}$, 2 from $m^{M(\tilde{p})}$, 3 from $k^{M(\tilde{p})}$ and 4 from $l^{M(\tilde{p})}$.

We extend these ideas a little further. Suppose that $\{Z_i\}_{i<\omega}$ is uniformly computable from a low set $Z$ and represents a uniformly low set of degrees $Z$. Let $G$ be 1-generic relative to $Z$ and computable from $Z'$. Let $M(\tilde{p})$ be the model of arithmetic coded below $G$. Consider the sequence $\{Z_i \oplus G_{2i}\}_{i<\omega}$. This is an anti-chain, bounded by $G \oplus Z$. As $(G \oplus Z)' = G \oplus Z' \leq_T Z'$, it follows that $\{Z_i \oplus G_{2i}\}_{i<\omega}$ is a uniformly low anti-chain, bounded by a low degree and hence as well definable with parameters. It follows that the function mapping $i^M$ to $d_T(Z_i)$ is also definable.

**Definition 2.** Let $Z$ be a set of degrees, represented by a sequence $\{Z_i\}_{i<\omega}$. We say that $\tilde{p}$ define an indexing of $Z$ via the sequence $\{Z_i\}_{i<\omega}$ if they define a model of arithmetic $M$ and a function $\varphi : \mathbb{N}^M \rightarrow \mathcal{D}$ such that $\varphi(i^M) = d_T(Z_i)$.

The discussion above allows us to phrase the local coding theorem in a slightly stronger form.

**Theorem 2.1** (Slaman and Woodin [9]). For every uniformly low set of degrees $Z$ and uniformly low sequence $\{Z_i\}_{i<\omega}$ representing $Z$ there are finitely many parameters $\tilde{p} \in D_T(\leq 0')$ that define an indexing of $Z$ via $\{Z_i\}_{i<\omega}$.

Slaman and Woodin [9] prove that the set of computably enumerable degrees $\mathcal{R}$ is definable with finitely many parameters from $D_T(\leq 0')$. The proof uses an idea from Welch [10]. Let $\{W_e\}_{e<\omega}$ be the standard listing of all c.e. sets and let $K = \{(e,x) \mid x \in W_e\}$. By the Sacks Splitting Theorem [3] there are two low sets $A$ and $B$ such that $A \cup B = K$. Let $A = \bigoplus_{e<\omega} A_e$ and $B = \bigoplus_{e<\omega} B_e$. Then $A = \{d_T(A_e) \mid e < \omega\}$ and $B = \{d_T(B_e) \mid e < \omega\}$ are uniformly low sets of degrees and for every $e$ we have that $d_T(W_e) = d_T(A_e) \vee d_T(B_e)$. If $\tilde{p}$ and $\tilde{q}$ are parameters that code standard models of arithmetic $M(\tilde{p})$ and $M(\tilde{q})$ then the mapping $n^{M(\tilde{p})} \rightarrow n^{M(\tilde{q})}$ is definable from $\tilde{p}$ and $\tilde{q}$. Thus by extending the number of parameters we use, we can assume that any finite number of indexings use the same coded model of arithmetic. Thus we can find a finite list of parameters $\tilde{p}$ which code a model $M$, an indexing of the sequence $\{A_e\}_{e<\omega}$ and an indexing of the sequence $\{B_e\}_{e<\omega}$. We combine these to obtain an indexing of the c.e. degrees via the sequence $\{W_e\}_{e<\omega}$. We will call this an indexing of the c.e. degrees. In line with Theorem 2.1 we restate the definability of the c.e. degrees from parameters as follows.

**Theorem 2.2** (Slaman and Woodin [9]). There are finitely many parameters in $D_T(\leq 0')$ that define an indexing of the c.e. degrees.
This statement is stronger than the definability with parameters of the c.e. degrees. Let $A$ be a set of natural numbers with the following closure property: if $e \in A$ and $W_e \equiv_T W_j$ then $j \in A$. $A$ induces a set of c.e. degrees $A = \{d_T(W_e) \mid e \in A\}$. It follows from Theorem 2.2 that if $A$ is arithmetical then $A$ is definable in $D_T(\leq 0')$ with parameters. To see this, let $\vec{p}$ be parameters that define a model of arithmetic $M$ and an indexing $\varphi$ of the c.e. degrees. We use the arithmetic definition of $A$, effectively translated in the language of degree theory, to define the set $A^M$ of the degrees $e^M$ such that $e \in A$ and then observe that $A = \varphi(A^M)$ is definable from $\vec{p}$.

3. Biinterpretability with parameters

Consider the sequence $\{X_e\}_{e<\omega}$, where

- If $\{e\}^{0'}$ is a total $\{0,1\}$-valued function then $X_e$ is the set with characteristic function $\{e\}^{0'}$.
- $X_e = \emptyset$ otherwise.

Then $\mathcal{X} = \{d_T(X_e) \mid e < \omega\}$ is the set of all $\Delta^0_2$ degrees. An indexing of the $\Delta^0_2$ degrees via the sequence $\{X_e\}_{e<\omega}$ will be called simply an indexing of the $\Delta^0_2$ degrees.

We will say that $D_T(\leq 0')$ is biinterpretable with first order arithmetic if there is a definable indexing of the $\Delta^0_2$ degrees. We use ideas similar to the ones described in the previous section to show that there are finitely many parameters in $D_T(\leq 0')$ that define such an indexing, i.e. that $D_T(\leq 0')$ is biinterpretable with first order arithmetic modulo the use of finitely many parameters.

We will state two technical theorems and leave their proofs for the last two sections of this paper. The first one will allow us to reduce our task to finding an indexing of the low $\Delta^0_2$ degrees.

**Theorem 3.1.** Let $y < 0'$. There are low degrees $x_1, x_2, x_3$ and $x_4$ such that

$$y = (x_1 \lor x_2) \land (x_3 \lor x_4).$$

The second technical result shows that there is a uniformly low set of degrees $Z$, bounded by a low degree, such that every low Turing degree has a unique position with respect to the elements of $Z$ and the c.e. degrees.

**Theorem 3.2.** There exists a uniformly low set of Turing degrees $Z$, bounded by a low degree $z < 0'$, such that if $x, y < 0'$, $x' = 0'$ and $y \nleq x$ then there are c.e. degrees $a_i$ and $\Delta^0_2$ Turing degrees $c_i$, $b_i$, $g_i$ for $i = 1, 2$ such that:

1. $b_i$ and $c_i$ are elements of $Z$.
2. $g_i$ is the least element below $a_i$ which joins $b_i$ above $c_i$.
3. $x \leq g_1 \lor g_2$.
4. $y \nleq g_1 \lor g_2$.

We state and prove our main result.

**Theorem 3.3.** There are parameters $\vec{p}$ in $D_T(\leq 0')$ which define an indexing of the $\Delta^0_2$ degrees.

**Proof.** Let $\{Z_i\}_{i<\omega}$ be the uniformly low sequence of sets, representing the uniformly low set of degrees $Z$ obtained in Theorem 3.2. By Theorem 2.1 and Theorem
2.2 we can fix finitely many $\Delta^0_2$ parameters $\vec{p}$ which code a model $\mathcal{M}$, an indexing $\varphi_{\mathcal{Z}}$ of $\mathcal{Z}$ via $\{Z_i\}_{i<\omega}$ and an indexing $\varphi_{\mathcal{R}}$ of the c.e. degrees:

$$\varphi_{\mathcal{Z}}(i^M) = d_T(Z_i) \quad \text{and} \quad \varphi_{\mathcal{R}}(i^M) = d_T(W_i).$$

We will show that the function $\varphi : \mathbb{N}^M \to D_T(\leq 0')$, given by $\varphi(i^M) = d_T(X_i)$ is definable in $D_T(\leq 0')$ from the parameters $\vec{p}$.

Fix a $\Delta^0_2$ degree $i$. The property $i \in \mathbb{N}^M$ is definable from the parameters $\vec{p}$, so we can assume that $i = i^M$ for a fixed natural number $i$. We have different cases depending on the nature of the number $i$. We have an effective way to translate an arithmetic statement about $i$, say $\psi(i)$, into a degree theoretic statement $\psi(i, \vec{p})$ so that:

$$\mathbb{N} \models \psi(i) \quad \text{if and only if} \quad D_T(\leq 0') \models \psi(i, \vec{p}).$$

We will say that $\psi(i)$ is true in $\mathcal{M}$ to mean that $\psi(i, \vec{p})$ is true in $D_T(\leq 0')$.

**Case 1:** Suppose that in $\mathcal{M}$ the following statement is true:

“$\{i\}^W$ is not a $\{0,1\}$-valued total function”.

In this case $\varphi(i) = 0$.

**Case 2:** Suppose that in $\mathcal{M}$ the following statement is true:

“$\{i\}^W$ is the characteristic function of a low set.”

In this case we use Theorem 3.2. Note that as $\{Z_i\}_{i<\omega}$ is a sequence uniformly computable from $0'$, we can fix a total computable function $g$, such that $Z_c = \{g(e)\}^W$. We can define $\varphi(i)$ as the largest degree $x$ with the following property: For every list of degrees $a_1, a_2, b_1, b_2, c_1, c_2, g_1$ and $g_2$ if:

- $g_1$ is the least element below $a_1$ which joins $b_1$ above $c_1$;
- $g_2$ is the least element below $a_2$ which joins $b_2$ above $c_2$;
- There are elements $e_{a_1}, e_{b_1}$ and $e_{c_1}$ such that $\varphi_{\mathcal{R}}(e_{a_1}^M) = a_1$, $\varphi_{\mathcal{Z}}(e_{b_1}^M) = b_1$ and $\varphi_{\mathcal{Z}}(e_{c_1}^M) = c_1$;
- There are elements $e_{a_2}, e_{b_2}$ and $e_{c_2}$ such that $\varphi_{\mathcal{R}}(e_{a_2}^M) = a_2$, $\varphi_{\mathcal{Z}}(e_{b_2}^M) = b_2$ and $\varphi_{\mathcal{Z}}(e_{c_2}^M) = c_2$;
- In $\mathcal{M}$ the following statement is true:

\[ d_T(\{i\}^W) \text{ is bounded by the join of the least Turing degree below} \]
\[ d_T(W_{e_{a_1}}) \text{ which joins} \]
\[ d_T(\{g(e_{b_1})\}^W) \text{ above} \]
\[ d_T(\{g(e_{c_1})\}^W) \text{ and} \]
\[ \text{the least Turing degree below} \]
\[ d_T(W_{e_{a_2}}) \text{ which joins} \]
\[ d_T(\{g(e_{b_2})\}^W) \]
\[ \text{above} \]
\[ d_T(\{g(e_{c_2})\}^W). \]

then $x \leq g_1 \lor g_2$.

**Case 3:** Suppose that in $\mathcal{M}$ the following statement is true:

“$\{i\}^W$ is the characteristic function of a non low and incomplete set.”

In this case we apply Theorem 3.1 and the previous case. Let $e_1, e_2, e_3, e_4$ be natural numbers such that in $\mathcal{M}$ the following statement is true:

For each $j = 1, 2, 3, 4$ the function $\{e_j\}^W$ is the characteristic function of a low set and
\[ d_T(\{i\}^W) = (d_T(\{e_1\}^W) \lor d_T(\{e_2\}^W)) \land (d_T(\{e_3\}^W) \lor d_T(\{e_4\}^W)). \]

Under the previous case we have already shown that $\varphi(e_j^M)$ is definable with parameters $\vec{p}$. We define $\varphi(i)$ as $(\varphi(e_1^M) \lor \varphi(e_2^M)) \land (\varphi(e_3^M) \lor \varphi(e_4^M))$. 
Case 4: Finally, \( \varphi(i) = 0' \) if in \( \mathcal{M} \) the following statement is true
\[ \{i\}^{\mathcal{M}} \text{ is total, and computes } 0'. \]

\[ \square \]

4. Applications

Biinterpretability with parameters has many consequences for \( \mathcal{D}_T(\leq 0') \) and its automorphism group. In this section we state and prove them.

**Definition 3.** An automorphism base of a structure \( \mathcal{A} \) with domain \( A \) is a set \( B \subseteq A \), such that for every pair of automorphisms \( \pi_1 \) and \( \pi_2 \) of \( \mathcal{A} \) if \( \pi_1 \) and \( \pi_2 \) agree on all elements in \( B \) then \( \pi_1 = \pi_2 \).

The relationship between an indexing of the \( \Delta^0_2 \) degrees and automorphism bases is given by the following.

**Theorem 4.1.** If \( \bar{\pi} \) are parameters that define an indexing of the \( \Delta^0_2 \) degrees then \( \bar{\pi} \) is an automorphism base for \( \mathcal{D}_T(\leq 0') \).

**Proof.** Consider an automorphism \( \pi \) of \( \mathcal{D}_T(\leq 0') \). And let \( \bar{\pi} \) be the \( \Delta^0_2 \) parameters which code a model of arithmetic \( \mathcal{M} \) and an indexing \( \varphi \) of the \( \Delta^0_2 \) degrees. Then \( \pi(\bar{\pi}) \) are degrees which also code a model of arithmetic \( \mathcal{M}_\pi \) and a function:
\[ \varphi_\pi : \mathbb{N}_{\mathcal{M}_\pi} \to \Delta^0_2 \] which is onto. As every element in the domain of the model of arithmetic is definable from the parameters that code it, it follows that for every natural number \( e \) we have that \( \pi(e^{\mathcal{M}}) = e^{\mathcal{M}_\pi} \). This allows us to conclude that the image of \( \bar{\pi} \) determines the image of every other \( \Delta^0_2 \) degree, as for every natural number \( e \) we have that \( \pi(d_T(\{e\}^{\emptyset'})) = \pi(\varphi(e^{\mathcal{M}})) = \varphi_\pi(\pi(e^{\mathcal{M}})) = \varphi_\pi(e^{\mathcal{M}_\pi}) \). In particular if two automorphism agree on the elements of \( \bar{\pi} \) then they are identical. \[ \square \]

**Definition 4.** Let \( \pi \) be a function on degrees. A presentation of \( \pi \) is a function \( \Pi \) on sets, such that for every set \( X \) we have that \( d_T(\Pi(X)) = \pi(d_T(X)) \).

**Corollary 4.2.** \( \mathcal{D}_T(\leq 0') \) has a finite automorphism base. The automorphism group of \( \mathcal{D}_T(\leq 0') \) is countable and every automorphism of \( \mathcal{D}_T(\leq 0') \) has an arithmetically definable presentation.

**Proof.** By Theorem 3.3 there are finitely many parameters \( \bar{\pi} \) which define an indexing of the \( \Delta^0_2 \) degrees. By Theorem 4.1 they are an automorphism base for \( \mathcal{D}_T(\leq 0') \). There are only countably many choices for the image of \( \bar{\pi} \) and hence only countably many possible automorphisms.

The procedure, described in the proof of Theorem 4.1, for determining \( \pi(d_T(\{e\}^{\emptyset'})) \) from the image of \( \bar{\pi} \) is arithmetically definable and therefore gives us an arithmetic presentation of \( \pi \). \[ \square \]

**Definition 5.** We say that a structure is atomic if the complete type of every tuple is axiomatized by a single formula.

**Theorem 4.3.** \( \mathcal{D}_T(\leq 0') \) is atomic.

**Proof.** We show that for every tuple \( \bar{x} \) there is a formula \( \chi_{\bar{x}} \) which determines the type of \( \bar{x} \) in \( \mathcal{D}_T(\leq 0') \). Let \( \bar{\pi} \) be parameters that code a model of arithmetic \( \mathcal{M} \) and an indexing \( \varphi \) of the \( \Delta^0_2 \) degrees. The indexing \( \varphi \) is far from injective, but we can use it to define an injection \( \theta : \mathcal{N}^{\mathcal{M}} \to \mathcal{D}_T(\leq 0') \). In arithmetic we can define
a function \( f : \mathbb{N} \to \mathbb{N} \) where \( f(n) \) is the least index of a \( \Delta^0_2 \) set which is not Turing equivalent to any of the sets with indices \( f(m) \) for \( m < n \). Let \( \theta(e^M) = \varphi(f(e^M)) \).

Now \( \vec{x} = (x_1, \ldots, x_k) \) is the tuple \((\theta(e^M_1), \ldots, \theta(e^M_k))\) for a unique tuple \((e_1, \ldots, e_k)\).

We define the formula \( \chi_{\vec{x}} \) to express exactly this relationship: there are parameters \( \vec{p} \) which code a standard model of arithmetic \( \mathcal{M} \), such that the relation \( \theta \) coded by \( \vec{p} \) using the definition that can be derived by combining the proof of Theorem 3.3 and the description of the function \( f \) above is a bijection \( \theta : \mathbb{N}^M \to \Delta^0_2 \), such that:

- For every \( i \) and \( j \) the model \( \mathcal{M} \) satisfies the translation of the arithmetic formula \( \{f(i)\}^\theta \leq_T \{f(j)\}^\theta \) if and only if \( \theta(i^M) \leq \theta(j^M) \).
- For every \( j \leq k \), \( \theta(e^M_j) = x_j \).

To guarantee that \( \mathcal{M} \) is a standard model of arithmetic we ask that for every other set of parameters \( \vec{q} \) which also code a model of arithmetic \( \mathcal{N} \) and a bijection \( \psi : \mathbb{N}^N \to \Delta^0_2 \) with the same property, we have that the following is true. Let \( \lambda : \mathbb{N}^M \to \mathbb{N}^N \) be the bijection defined by \( \lambda(e^M) = \psi^{-1}(\theta(e^M)) \). We require that the image of the interval \([0^M, n^M] \) into \( \mathcal{N} \) under \( \lambda \) is bounded in \( \mathcal{N} \). In other words we ask that for every element \( n^M \in \mathbb{N}^M \) there is an element \( b \in \mathbb{N}^N \) such that for every \( m \leq n \), the element \( \lambda(m^M) \leq_N b \).

Now if \( \vec{x} \) and \( \vec{y} \) satisfy the same formula \( \chi_{\vec{x}} \) then \( \vec{x} \) and \( \vec{y} \) are automorphic. Indeed, let \( \mathcal{M}_x, \theta_x \) be the model of arithmetic and bijection witnessing \( \chi_{\vec{x}}(\vec{x}) \) and \( \mathcal{M}_y, \theta_y \) be the model of arithmetic and bijection witnessing \( \chi_{\vec{x}}(\vec{y}) \). The automorphism \( \pi \) works as follows: on a degree \( \vec{z} \), it finds the natural number in the first model \( e^{M_x} \) such that \( \theta_x(e^{M_x}) = \vec{z} \) and then maps \( \vec{z} \) to \( \theta_y(e^{M_y}) \). As \( \theta_x \) and \( \theta_y \) are bijections, it follows that \( \pi \) is a bijection and \( \pi(\vec{x}) = \vec{y} \). Furthermore, for all pairs of degrees \( \vec{z}_1 = \theta_x^{-1}(e^M_1) \) and \( \vec{z}_2 = \theta_x^{-1}(e^M_2) \) we have that \( \vec{z}_1 \leq \vec{z}_2 \) if and only if \( \{f(e_1)\}^\theta \leq_T \{f(e_2)\}^\theta \) if and only if \( \pi(\vec{z}_1) \leq \pi(\vec{z}_2) \), so \( \pi \) is an automorphism.

We next consider consequences of biinterpretability with parameters to first order definability in \( D_T(\leq 0') \).

**Definition 6.** Let \( R \subseteq \mathbb{N}^k \).

1. \( R \) is degree invariant if whenever \( (e_1, \ldots, e_k) \in R \) and \( \{e_1\}^\theta \equiv_T \{i_1\}^\theta, \ldots, \{e_k\}^\theta \equiv_T \{i_k\}^\theta \) then \( (i_1, \ldots, i_k) \in R \).
2. If \( R \) is degree invariant then \( R \) induces a relation \( \mathcal{R} \) on \( D_T(\leq 0') \) defined by \( (x_1, \ldots, x_k) \in \mathcal{R} \) if and only if \( x_1 = d_T(\{e_1\}^\theta), \ldots, x_k = d_T(\{e_k\}^\theta) \) and \( (e_1, \ldots, e_k) \in R \).

**Theorem 4.4.**

1. Every relation on \( D_T(\leq 0') \) that is induced by an arithmetically definable degree invariant relation is definable in \( D_T(\leq 0') \) with finitely many \( \Delta^0_2 \) parameters.
2. Every relation on \( D_T(\leq 0') \) that is induced by an arithmetically definable degree invariant relation and which is invariant under automorphisms is definable in \( D_T(\leq 0') \).

**Proof.** Let \( R \subseteq \mathbb{N}^k \) be an arithmetically definable degree invariant set and \( \mathcal{R} \) be the induced relation on \( D_T(\leq 0') \). Let \( \vec{p} \) be the parameters which code a model of arithmetic \( \mathcal{M} \) and an indexing of the \( \Delta^0_2 \) degrees \( \varphi \) as in Theorem 3.3. Let \( \chi(x_1, \ldots, x_k) \) be the formula in arithmetic which defines \( \mathcal{R} \). Then \( \mathcal{R} \) is definable in \( D_T(\leq 0') \) with parameters \( \vec{p} \) by the following formula: \( (x_1, \ldots, x_k) \in \mathcal{R} \) if and
only if there are elements \(e_1^\mathcal{M}, \ldots, e_k^\mathcal{M}\) in the domain of the model \(\mathcal{M}\) such that 
\[x_i = \varphi(e_i^\mathcal{M})\] for \(i \leq k\) and such that \(\chi(e_1, \ldots, e_k)\) is true in \(\mathcal{M}\).

If \(\mathcal{R}\) is furthermore invariant under automorphisms then we use the method from the proof of Theorem 4.3. Recall that \(f\) is a function with the property that \(f(n)\) is the least index of a \(\Delta^0_2\) set which is not Turing equivalent to any of the sets with indices \(f(m)\) for \(m < n\). We define \((x_1, \ldots, x_k) \in \mathcal{R}\) if and only if there are parameters \(\vec{p}\) which code a standard model of arithmetic \(\mathcal{M}\) and a bijection \(\theta : \mathbb{N}^\mathcal{M} \to \Delta^0_2\), such that:

- For every \(i\) and \(j\) the model \(\mathcal{M}\) satisfies the translation of the arithmetic formula \(\{f(i)\}^{\theta} \leq_T \{f(j)\}^{\theta}\) if and only if \(\theta(i^\mathcal{M}) \leq \theta(j^\mathcal{M})\).
- There are elements \(e_1^\mathcal{M}, \ldots, e_k^\mathcal{M}\) in the domain of the model \(\mathcal{M}\) such that 
  \[\chi(e_1, \ldots, e_k)\] for \(i \leq k\) and such that \(\chi(f(e_1), \ldots, f(e_k))\) is true in \(\mathcal{M}\).

Let \(\vec{p}\) be the parameters that code a model of arithmetic and a bijective indexing of the \(\Delta^0_2\) degrees \(\theta\). If \((x_1, \ldots, x_k) \in \mathcal{R}\) then the parameters \(\vec{p}\) witness that \((x_1, \ldots, x_k)\) satisfy this formula. If on the other hand \((x_1, \ldots, x_k)\) satisfy the formula above as witnessed by parameters \(\vec{q}\) then there is a tuple \((y_1, \ldots, y_k)\) that satisfies the formula as witnessed by \(\vec{p}\). As \(\vec{p}\) defines an indexing of the \(\Delta^0_2\) degrees, we have that \((y_1, \ldots, y_k) \in \mathcal{R}\). Furthermore \((y_1, \ldots, y_k)\) is automorphic to \((x_1, \ldots, x_k)\). As \(\mathcal{R}\) is invariant under automorphisms, it follows that \((x_1, \ldots, x_k) \in \mathcal{R}\).

**Corollary 4.5.** \(\mathcal{D}_T(\leq 0')\) is rigid if and only if it is biinterpretable with first order arithmetic.

**Proof.** If \(\mathcal{D}_T(\leq 0')\) has no nontrivial automorphisms then every relation on \(\mathcal{D}_T(\leq 0')\) is invariant under automorphisms. So by Theorem 4.4 every relation \(\mathcal{R}\), induced by a degree invariant relation \(\mathcal{R}\) definable in first order arithmetic, is definable in \(\mathcal{D}_T(\leq 0')\). In particular, consider the parameters \(\vec{p} = (p_1, \ldots, p_k)\) which define an indexing of the \(\Delta^0_2\) degrees. The relation \(\mathcal{R} = \{(e_1, \ldots, e_k) \mid (\forall i \leq k)\{e_i\}^{\theta} \in p_i\}\) is definable and induces \(\mathcal{R} = \{\vec{p}\}\). It follows that the indexing \(\varphi\) is definable.

If there is a definable indexing \(\varphi\) of the \(\Delta^0_2\) degrees then by Theorem 4.1 the empty set is an automorphism base of \(\mathcal{D}_T(\leq 0')\). In other words there can be only one automorphism, the identity. It follows that \(\mathcal{D}_T(\leq 0')\) is rigid.

The applications listed in this section give a clearer path towards understanding the structure of the \(\Delta^0_2\) Turing degrees, revealing deep connections between definability, rigidity and biinterpretability. By analyzing the complexity of the formula \(\chi\) for tuples of length 1 in Theorem 4.3 we obtain a number \(b\) such that:

1. There is a sequence of formulas with one free variable and of fixed quantifier complexity \(b\): \(\varphi_1(x), \varphi_2(x), \ldots\) such that:
   - Every \(\Delta^0_2\) degree \(d\) satisfies at least one such formula.
   - If any one of these formulas is satisfied by two different \(\Delta^0_2\) degrees then \(\mathcal{D}_T(\leq 0')\) has a nontrivial automorphism.
2. If \(d\) is a fixed point of an automorphism of the structure \(\mathcal{D}_T(\leq 0')\) then \(d\) is first order definable in \(\mathcal{D}_T(\leq 0')\) by a formula with complexity \(b\).
3. The following two statements are equivalent:
   - No intermediate degree is definable in \(\mathcal{D}_T(\leq 0')\).
   - No intermediate degree is definable in \(\mathcal{D}_T(\leq 0')\) by a formula with quantifier complexity \(b\).
The equivalent statements listed in (3) above imply rigidity for \( D_T(\leq 0') \) in a very strong way. They allow us, on the other hand, to formulate a clear technical route to settling the automorphism problem for the structure \( D_T(\leq 0') \):

**Question 1.** Let \( b > 0 \) be a natural number. Given a formula \( \varphi(x) \) of quantifier complexity \( b \) and an intermediate Turing degree \( d \), such that \( \varphi(d) \) is true in \( D_T(\leq 0') \), does there exist an intermediate degree \( d^* \neq d \) such that \( \varphi(d^*) \) is also true of \( D_T(\leq 0') \).

This question has an easy answer when \( b = 1 \). Any existential formula true of an intermediate \( \Delta^0_2 \) degree \( d \) can be realized by a \( \Delta^0_2 \) degree \( g \) that is 1-generic relative to \( d \). A negative answer to this question at any level \( b \) would provide us with a definable intermediate degree. The most interesting \( b \) is the one from the discussion above: a positive answer for all formulas at that level of complexity would show that every intermediate point has a nontrivial orbit; in the other extreme, there are finitely many formulas of that level, such that a negative answer for all of them would imply rigidity for \( D_T(\leq 0') \) and, by Slaman and Soskova [8], rigidity for the global structure of the enumeration degrees.

5. Low degrees determine all \( \Delta^0_2 \) degrees

This section is devoted to the proof of Theorem 3.1. We will prove that every incomplete \( \Delta^0_2 \) degree \( y \) can be represented as \( (x_1 \lor x_2) \land (x_3 \lor x_4) \), where \( x_1, \ldots, x_4 \) are low degrees.

We start with a definition of a functional \( C \) which maps a pair of sets \( Y \) and \( G \) to a set \( C(Y, G) \) as follows. Let \( G = G_{\text{even}} \oplus G_{\text{odd}} \):

\[
C(Y, G)(n) = \begin{cases} 
G_{\text{even}}(n - m) & \text{if } G_{\text{odd}}(n) = 0 \text{ and } |G_{\text{odd}} \upharpoonright n| = m \\
Y(m) & \text{if } G_{\text{odd}}(n) = 1 \text{ and } n \text{ is the } m\text{-th element of } G_{\text{odd}}.
\end{cases}
\]

Note that \( G \oplus Y \equiv_T G_{\text{odd}} \oplus C(Y, G) \). We furthermore show the following property:

**Lemma 5.1.** If \( G \) is 1-generic relative to \( Y \) then so is \( C(Y, G) \).

**Proof.** Suppose that \( W \) is c.e. in \( Y \) and \( W \) is dense in \( C(Y, G) \), i.e every initial segment of \( C(Y, G) \) has an extension in \( W \). We need to show that \( C(Y, G) \) meets \( W \). We extend the definition of \( C \) so that it can have a finite binary string as a second parameter.

\[
C(Y, \tau)(n) = \begin{cases} 
\tau_{\text{even}}(n - m) & \text{if } \tau_{\text{odd}}(n) = 0 \text{ and } |\tau_{\text{odd}} \upharpoonright n| = m \\
Y(m) & \text{if } \tau_{\text{odd}}(n) = 1 \text{ and } n \text{ is the } m\text{-th element of } \tau_{\text{odd}}; \\
\uparrow & \text{if } \tau_{\text{odd}}(n) \uparrow.
\end{cases}
\]

Consider the c.e. set \( W^* = \{ \sigma \oplus \tau \mid C(Y, \sigma \oplus \tau) \in W \} \). This set is dense in \( G \). Indeed, fix \( \sigma \oplus \tau \preceq G \). Then \( C(Y, \sigma \oplus \tau) \) is an initial segment of \( C(Y, G) \), hence there is an extension of \( C(Y, \sigma \oplus \tau) \), say \( C(Y, (\sigma \oplus \tau) \gamma) \) in \( W \). We can represent \( C(Y, (\sigma \oplus \tau) \gamma) \) as \( C(Y, (\sigma \gamma) \oplus (\tau 00\ldots 0)) \), where the number of zeros added at the end of \( \tau \) is the length of \( \gamma \). Thus \( (\sigma \gamma) \oplus (\tau 00\ldots 0) \) is an extension of \( \sigma \oplus \tau \) in \( W^* \). As \( G \) is 1-generic relative to \( Y \), it follows that some initial segment \( \sigma \oplus \tau \) of \( G \) is in \( W^* \), but then \( C(Y, \sigma \oplus \tau) \) is an initial segment of \( C(Y, G) \) in \( W \). \( \square \)
Let \( y < 0' \) be a fixed incomplete \( \Delta^0_2 \) Turing degree and fix \( Y \in y \). We will use the well known fact that every non computable c.e. set computes a 1-generic set, hence relativizing if \( Z \) is a set that is c.e. in \( Y \) and \( Y' \) \( Z \) computes a 1-generic relative to \( Y \). See Downey and Hirschfeldt [1] for a proof of this fact. As \( 0' \) is c.e. relative to \( Y \) and \( Y' \) \( Y \) there is a set \( G \leq_T 0' \) which is 1-generic relative to \( Y \). We split \( G \) into odd and even bits: \( G = G_0 \oplus G_1 \). Then \( G_0 \) and \( G_1 \) are mutually generic relative to \( Y \) and so \( G_0 \oplus Y \) and \( G_1 \oplus Y \) form a minimal pair above \( Y \). We set \( X_1 \) to be the odd bits in \( G_0 \) and \( X_2 = C(Y,G_0) \). Similarly \( X_3 \) is the odd bits in \( G_1 \) and \( X_4 = C(Y,G_1) \). The sets \( X_i, i = 1 \ldots 4 \) are 1-generic and hence low. Now we have the required property:

\[
y = (d_T(X_1) \lor d_T(X_2)) \land (d_T(X_3) \lor d_T(X_4)).
\]

6. A uniformly low set of Turing degrees

This section is devoted to the proof of Theorem 3.2. We wish to prove that there is a uniformly low set \( Z \) of Turing degrees, bounded by a low \( z < 0' \), such that if \( x,y \) are \( \Delta^0_2 \) degrees, \( x \) is low and \( y \not\leq x \), then there are \( g_i < 0', \) c.e. degrees \( a_i \) and \( \Delta^0_2 \) Turing degrees \( c_i, b_i \) for \( i = 1,2 \) such that:

1. \( b_i \) and \( c_i \) are elements of \( Z \).
2. \( g_i \) is the least element below \( a_i \) which joins \( b_i \) above \( c_i \).
3. \( x \leq g_1 \lor g_2 \).
4. \( y \not\leq g_1 \lor g_2 \).

The first step in our proof is to design a construction which takes as parameters approximations to sets \( X \) and \( Y \) and produces approximations to sets \( G_i, A_i, B_i, C_i \). If \( X \) is low, as witnessed by the given approximation, and \( Y \) is \( \Delta^0_2 \) and not computable from \( X \) then the sets \( G_i, A_i, B_i, C_i \) will have the necessary structural properties. Once we have such a construction we will run it simultaneously relative to all possible parameters and interweave that with lowness requirements to obtain the uniformly low set \( Z \).

We start with the description of the construction. Let \( \{X[s]\}_{s<\omega} \) and \( \{Y[s]\}_{s<\omega} \) be uniformly computable sequences of finite sets.

**Definition 7.** A \( \Delta^0_2 \) approximation \( \{X[s]\}_{s<\omega} \) to a set \( X \) is called a low approximation, if it has the following property: for every Turing functional \( \Xi \) and natural number \( x \) if there are infinitely many stages \( s \) such that \( \Xi^X(x)[s] \downarrow \) then \( \Xi^X(x) \downarrow \).

We construct c.e sets \( A_i, \Delta^0_2 \) sets \( B_i \) and \( C_i \) and uniformly computable sequences \( \{G_i[s]\}_{s<\omega} \) so that if \( \{X[s]\}_{s<\omega} \) is a low \( \Delta^0_2 \) approximation to a set \( X \) and \( \{Y[s]\}_{s<\omega} \) is a \( \Delta^0_2 \) approximation to a set \( Y \not\leq_T X \) then \( \{G_i[s]\}_{s<\omega} \) are \( \Delta^0_2 \) approximations to sets \( G_i \), where \( i = 1,2 \), such that the following requirements are satisfied:

1. \( A_i \): There is a Turing functional \( A_i \) such that \( G_i = \Lambda^A_i \).
2. \( \Gamma_i \): There is a Turing functional \( \Gamma_i \) such that \( C_i = \Gamma^B_i \).
3. \( \Omega : \) There is a Turing functional \( \Omega \) such that \( X = \Omega^{G_1,G_2} \).
4. Let \( (\Theta_{e}, \Phi_{e})_{e<\omega} \) be a listing of all pairs of Turing functionals. For every \( e \) we have the requirement:
   \( \mathcal{R}_{e} : \) If \( \Theta^\Phi_{e} \) \( B_i = C_i \) then there is a functional \( \Delta_{e} \) such that \( G_i = \Delta^\Phi_{e} \).
5. Let \( (\Psi_{e})_{e<\omega} \) be a listing of all Turing functionals. For every \( e \) we have the requirement:
   \( \mathcal{Q}_{e} : \Psi_{e}^{G_1,G_2} \neq Y \).
6.1. Description of strategies. Suppose that \( \{X[s]\}_{s<\omega} \) is a low \( \Delta^0_2 \) approximation to a set \( X \) and \( \{Y[s]\}_{s<\omega} \) is a \( \Delta^0_2 \) approximation to a set \( Y \nsubseteq_T X \). The construction is in stages. We start stage \( s \) by visiting the global strategies: \( \Lambda_0, \Lambda_1, \Gamma_0, \Gamma_1 \), and \( \Omega \).

The \( \Lambda \)-strategies. The \( \Lambda_i \)-strategy is a simple marker strategy. During the construction we dynamically assign markers \( \lambda_i(n) \) to every natural number \( n \). At the beginning of a stage we check \( \Lambda_i \) for errors. If \( \Lambda_i^{\alpha_i}(n) \uparrow \) then we define it to be equal to \( G_i(n) \) with use \( A_i \nmid \lambda_i(n) \), where \( \lambda_i(n) \) is a new fresh number. A fresh number is defined as follows: let \( N \) be the largest number mentioned so far in the construction; a fresh number is larger than \( 2N \). This way we not only use a new number every time, but also leave enough space between any two numbers that are used. Here we mean \( A \uparrow x \) to be the initial segment of \( A \) of length \( x + 1 \). During the construction, whenever we change the value of \( G_i(n) \), we will automatically also enumerate the marker \( \lambda_i(n) \) in \( A_i \), so that whenever \( \Lambda_i^{\alpha_i}(n) \) is defined, it is consistent with \( G_i(n) \). This will be the only reason that \( A_i \) changes. So, as long as \( G_i(m) \) changes only finitely many times for every \( m \leq n \) we will have that \( \Lambda^{\alpha_i}(n) \) is eventually defined.

The \( \Gamma \)-strategies. The strategy for the construction of \( \Gamma_i \) is slightly more complicated for the following reason. The \( \Delta^0_2 \) sets \( B_i \) and \( G_i \) can change many times. We cannot raise the value of the use of an element up every time this happens, or else we risk to make \( \Gamma_i \) partial. Instead to every element \( c \) we will attach two markers \( g_i(c) \) and \( \gamma_i(c) \). The marker \( g_i(c) \) will serve as the \( G_i \)-use of \( c \). It is usually set to \( 0 \), however an \( R_i \)-strategy can preemptively raise its value once during the construction. The marker \( \gamma_i(c) \) is the \( B_i \)-use of \( c \) and it is raised only if we observe a change in \( G_i \upharpoonright g_i(c) \) or a change in \( C_i \upharpoonright c \). The \( \Gamma_i \)-strategy only defines new computations, it never changes the approximation to any set. At stage \( s \) it ensures that \( \Gamma_i^{B_i,G_i}(c) \) is defined for all \( c < s \). The strategies for \( R_i \) requirements will ensure that \( \Gamma_i \) is always correct.

The \( \Omega \)-strategy. The strategy to define \( \Omega \) will once again be able to move the use up every time we see that some computation is undefined. It will follow from the construction that \( G_1 \) and \( G_2 \) almost always change only in response to \( X \) changes. We will maintain that any \( \Omega \) computation ever defined will use an initial segment of the oracle ending in \( 0 \). Thus enumerating the last bit of the use in the oracle set will invalidate the computation. At stage \( s \) we ensure that \( \Omega^{G_1,G_2}(n) \) is correctly defined for every \( x < s \), by changing the approximation to \( G_1 \) or \( G_2 \) if necessary to invalidate wrong computations. If \( \Omega^{G_1,G_2}(n) \uparrow \) we define a new computation for \( n \) with a fresh \( \Omega \)-use (also referred to as an \( \Omega \)-marker) \( \omega(n) \).

Now we turn to the two more difficult requirements \( R_i^e \) and \( Q_e \). We will handle them using a tree of strategies. We will denote \( R_i^e \) strategies by \( \alpha \) (or \( \alpha^i \)), and \( Q_e \) strategies by \( \beta \). We order the requirements by priority linearly as follows:

\[
R_0^1 < \cdots < R_{e}^1 < R_e^2 < Q_e < R_{e+1}^1 \ldots
\]

Strategies of level \( n \) in the tree are assigned the \( n \)-th requirement in the priority ordering. The branching in the tree is determined by the outcomes of the strategies. The lexicographical ordering of nodes in the tree induces a priority ordering of the strategies.

The \( R \)-strategies. Suppose that \( \alpha \) is working towards satisfying the requirement \( R_e^1 \). We will drop the subscript \( e \) in the discussion below. The strategy \( \alpha \) tries to
establish and preserve a difference between $\Theta_{\Phi^{A_1}}$ and $C_1$. At the same time it builds an operator $\Delta$, so that if no diagonalization is possible then $\Delta_{\Phi^{A_1}} = G_1$.

The strategy will have three possible outcomes: $f < L i < L w$. The outcome $f$ will signify that the strategy was successful in diagonalizing $\Theta_{\Phi^{A_1}}$ against $C_1$. In order to preserve this diagonalization at further stages, the strategy will need to preserve some initial segment of $\Phi_{\Phi^{A_1}}$. To do this, the strategy will have to set things up as follows.

When $\alpha$ is first visited after initialization it only records the stage in a parameter $s_0$. It waits until a stage $s_1$, when the higher priority $\Omega$-strategy has finished the definition of its functional on all numbers less than $s_0$ (i.e. every time $X \upharpoonright s_0$ changes, $\alpha$ starts its work from the beginning). Once this is done, $\alpha$ can assume that the use in the $\Omega$-computation of $s_0$, denoted by $\omega(s_0)$, is fixed and $G_1 \upharpoonright \omega(s_0)$, and hence $A_1 \upharpoonright \omega(s_0)$, do not change at further stages. The strategy $\alpha$ will be designed so that it does not change the sets $B_1$ or $C_1$ on numbers less than $s_1$.

This will be important to ensure the uniform lowness of all parameters $B_1$ and $C_1$. At stage $s_1$ the strategy selects a threshold $d_\alpha$ as the least number which is greater than $\omega(s_0)$ and not in $G_2$: $\omega(s_0) + 1$. Note that by the way that we select fresh numbers, $d_\alpha$ is not an $\omega$-marker and is smaller than the $\omega$-marker of $s_0 + 1$. Now $\alpha$ can preserve an initial segment of $G_1$ and of $A_1$ by enumerating the threshold $d_\alpha$ in $G_2$.

Next $\alpha$ inspects the greatest common initial segment of $\Theta_{\Phi^{A_1}}$ and $C_1$. Its length is called the length of agreement. It compares this length with the greatest one observed at a previous true stages. If the length of agreement is bounded then $\Theta_{\Phi^{A_1}} \neq C_1$. So, while the current length of agreement is smaller than any of the previous ones the strategy has outcome $w$.

If the length of agreement has grown, we say that the stage is expansionary. At expansionary stages the strategy first searches for a possibility for diagonalization, and if none is found progresses on the construction of $\Delta$. As these actions will be related we start by explaining the method by which $\alpha$ constructs the operator $\Delta$.

The strategy is equipped with an infinite computable set $\mathcal{C}$ of natural numbers, called the list of chits. The chits for different strategies are disjoint. The chits are used to define computations in $\Delta$. Whenever we want to define a new value for $\Delta^{\Phi^{A_1}}(n)$ we first assign a new chit $c(n) \notin C_1$ to the number $n$, wait until we see that $\Theta_{\Phi^{A_1}}(c(n))$ is defined and then use the same initial segment of $\Phi^{A_1}$ to define $\Delta^{\Phi^{A_1}}(n)$. So for every computation in $\Delta$, we have that if $\Delta^{\Phi^{A_1}}(n) \upharpoonright 1$ then there is some chit $c(n)$ and some finite binary string $\tau_B$ such that $\Theta_{\Phi^{A_1}, \tau_B}(c(n)) \upharpoonright 1 = 0$. We will say that the computation of $\Delta(n)$ is defined via the triple $(c(n), \tau_B, \tau_B)$. If $\Delta^{\Phi^{A_1}}(n)$ is defined and is not equal to $G_1(n)$, then we can diagonalize by preserving $G_1$ and $A_1$, ensuring $\tau_B \subseteq B_1$ and enumerating $c(n)$ into $C_1$.

In order to use this plan for diagonalization, we need to ensure that it is not in conflict with the strategy $\Gamma_1$, the strategy that is trying to prove that $C_1$ is computable from $G_1 \oplus B_1$ via $\Gamma_1$. For this reason the construction will be designed so that every computation for $c(n)$ in $\Gamma_1$ which has $B_1$-use smaller than $|\tau_B|$ assumes
the same value of $G_1(n)$, namely $\Delta^*_{c}(n)$. This is where the marker $g_1(c(n))$ will be used. When a new chit is assigned to $n$, it is selected to be a number that $\Gamma_1$ has not yet been defined on and its marker $g_1(c(n))$ is set to be larger than $n$. If $G_1(n)$ changes before this chit is used to define a computation in $\Delta$, we will cancel it and pick a new chit for $n$. Thus when we diagonalize via a chit $c(n)$, the computations in $\Gamma_1$, defined before $c(n)$ is used to define $\Delta$, are invalid on account of $G_1$. Any remaining valid computations in $\Gamma_1$ for $c(n)$ can be invalidated with a change in $B_1$ above the initial segment that needs to be preserved. This second change will be at a number that we denote as $b_{c(n)}$.

A final point to take into consideration is the impact of higher priority strategies. If a higher priority $R_1$ strategy $\alpha'$ acts to diagonalize and changes the approximation to $B_1$, this might not take into account the fact that a lower priority strategy $\alpha$ is counting on some $b_c \in B_1$ for a chit $c$ that it used to diagonalize. To remedy this we will maintain that $B_1(b_c) = C_1(c)$, i.e. $\alpha'$ will be responsible to change the value at $C_1(c)$ if necessary.

The actions of $\alpha$ at an expansionary stage are therefore as follows: First it examines the definition of $\Delta_{\alpha}$, searching for a number $n$ with $\Delta_{\alpha +1}(n) \neq G_1(n)$. If it finds such a number then it uses the chit $c(n)$ to diagonalize and has outcome $f$, until (if ever) it is injured by a higher priority strategy. If no such number is found, the strategy considers all numbers less than the current stage, defines chits for them and if possible new values for the $\Delta$. In this case it has outcome $i$.

The $Q$-strategies. Let $\beta$ be a $Q$-strategy working on the requirement $Q_c$. We drop the index $c$ in the discussion below. The strategy $\beta$ will have one outcome $f$. The strategy will cancel all lower priority strategies if it acts. In order to diagonalize the strategy will try to build a Turing functional $\Xi$ such that $\Xi^X = Y$. As $Y \not<_{T} X$ we know that this attempt will fail.

Just like the $R$-strategies, $\beta$ records the first stage it is visited after initialization in a parameter $s_0$ and waits until a stage $s_1$, such that $\Omega$ has correctly defined its functional on numbers $x < s_0$. At stage $s_1$ the strategy starts its real work. It monitors the length of agreement between $Ψ^{G_1,G_2}$ and $Y$. At expansionary stages it translates the computation of $Ψ^{G_1,G_2}(n)$ into a computation $\Xi^X(n)$ by using an initial segment of $X$ that is long enough, as to ensure that if $X$ does not change on this initial segment, then neither will $G_1$ or $G_2$. The idea is to argue that if the length of agreement between $Ψ^{G_1,G_2}$ and $Y$ is unbounded then, so is the length of agreement between $\Xi^X$ and $Y$. But since $X$ is low as witnessed by the given approximation, this would mean that $\Xi^X = Y$, contradicting $Y \not<_{T} X$. Thus the length of agreement will eventually remain bounded and the impact of this strategy on the whole construction would be finitary.

As $X$ is $\Delta^2_1$ we are faced with the following difficulty: a computation $\Xi^X(n)$ can be invalidated via a change in $X$. This results in a change in $G_1 \oplus G_2$ and possibly a new computation in $Ψ^{G_1,G_2}(n)$. Later on $X$ can revert to its old state, breaking the connection between computations in $Ψ$ and $\Xi$. To restore this connection, we must change $G_1$ and $G_2$ back to their old state as well.

The strategy keeps a list $P$ of promises. A promise has the form $(\tau_X, \tau_G)$ and means that we promise: if $\tau_X \leq X$ then $\tau_G \leq G_1 \oplus G_2$. During any visit (after stage $s_1$) the first action of the strategy is to ensure that all promises are kept by changing the approximation to $G_1 \oplus G_2$ according to the promise list and the current approximation to the set $X$. 
Next \( \beta \) examines the length of agreement between \( \Psi^{G_1,G_2} \) and \( Y \). If the stage is not expansionary, it does nothing further. If the stage is expansionary, the strategy defines \( \Xi \) on all elements \( x \) below the length of agreement, adding new entries to the list of promises.

6.2. Construction. We combine the ideas described above in a formal construction. We have global strategies: \( \Lambda_0, \Lambda_1, \Gamma_0, \Gamma_1 \) and \( \Omega \) and a tree \( T \preceq \{ f, i, w \}^\omega \) of \( R \) and \( Q \) strategies. Every strategy \( \alpha \) in the tree \( T \) is given higher priority than all its successors and all nodes to the right of it in \( T \).

To every strategy we attach parameters as follows:

- (1) \( \Lambda_i \) and \( \Omega \) have one parameter each: the functional that they are building. They also assign markers \( \lambda_i(n) \) and \( \omega(n) \) for all natural numbers \( n \).
- (2) \( \Gamma_i \), in addition to its functional, assigns two types of markers to all numbers \( n: g_i(n) \) tells us how much of \( G_i \) to use in a computation for \( n \) and \( \gamma_i(n) \) tells us how much of \( B_i \) to use in a computation for \( n \).
- (3) An \( R^G \) strategy \( \alpha \) has the following parameters: \( s_0(\alpha) \), the stage when \( \alpha \) started work after being initialized; \( s_1(\alpha) \) the stage after the last time \( \alpha \) is restarted; a threshold \( d_\alpha \), used to ensure that \( G_i \) and \( A_i \) can be restrained; a functional \( \Delta_\alpha \); numbers \( l_\alpha \) and \( l_\alpha \) recording the development of length of agreement between \( \Theta^G_{\alpha \downarrow} \) and \( C_i \); a list of chits \( C_\alpha \). In addition the strategy will dynamically assign chits \( c(n) \) to natural numbers \( n \) and record the stage at which this happened in \( s(c(n)) \). If \( c(n) \) is ever enumerated in \( C_i \), we attach to it one of its \( \Gamma_i \)-markers \( b_{c(n)} \).
- (4) A \( Q_c \) strategy \( \beta \) has the following parameters: \( s_0(\beta) \), the stage when \( \beta \) started work after being initialized; \( s_1(\beta) \) the stage after the last time \( \beta \) is restarted; a list of promises \( \mathcal{P}_\beta \); a functional \( \Xi_\beta \); numbers \( l_\beta \) and \( l_\beta \), recording the development of the length of agreement between \( \Psi^{G_1,G_2} \) and \( Y \). When \( \beta \) is initialized all parameters become undefined. When \( \beta \) is restarted all parameters except \( s_0(\beta) \) become undefined.

Construction:

At stage 0 we initialize all strategies and set \( A_i = B_i = C_i = G_i = \emptyset \). At stage \( s \geq 0 \) all parameters inherit their values from the previous stage unless they are explicitly modified during stage \( s \). We start stage \( s \) by visiting the global strategies:

**Step I:** \( \Omega \): Scan all \( x < s \).

- (1) If \( \Omega^{G_1,G_2}(x) \uparrow \) then let \( \omega(x) \) be a fresh number and set \( \Omega^{G_1,G_2}(x) = X(x) \) with use \( (G_1 \upharpoonright \omega(x)) \odot (G_2 \upharpoonright \omega(x)) \) and move to \( x + 1 \). Otherwise go to 2.
- (2) If \( \Omega^{G_1,G_2}(x) \downarrow \neq X(x) \) then enumerate \( \omega(x) \) in \( G_1 \) and enumerate \( \lambda_1(\omega(x)) \) in \( A_1 \). Go to step 1.

**Step II:** \( \Lambda_i \): Scan all elements \( n < s \). If \( \Lambda_i(n) \uparrow \) then let \( \lambda_i(n) \) be a fresh number and define \( \Lambda_i^A(n) = G_i(n) \) with use \( A_i \upharpoonright \lambda_i(n) \).

**Step III:** \( \Gamma_i \): Scan all \( c < s \). If \( \Gamma_i^B,G_i(c) \uparrow \) then first check if any of the following is true: \( \gamma_i(c) \uparrow \); there is a number \( d \leq c \) such \( G_i(d)[s-1] \neq C_i(d)[s] \); there is a number \( n < g_i(c) \), such that \( G_i(n)[s] \neq G_i(n)[s] \). If so, then let \( \gamma_i(c) \) be a fresh
number. Otherwise do not change the value of $\gamma_i(c)$. Set $\Gamma^{B_i,G_i}_i(c) = C_i(c)$ with use $(B_i \uparrow \gamma_i(c), G_i \uparrow g_i(c))$. If $g_i(s) \uparrow$, set $g_i(s) = 0$.

**Step IV: Construction of $\delta$:** We construct a finite path $\delta[s]$ in $T$. The path $\delta[s]$ is defined inductively. We set $\delta[s] \uparrow 0 = \emptyset$. Once we have constructed $\delta[s] \uparrow k$, we check if $k = s$ or if the strategy $\delta[s] \uparrow k$ ends the current stage. If so, then $\delta[s] = \delta[s] \uparrow k$. Otherwise the strategy produces an outcome $o$, and $\delta[s] \uparrow k + 1 = (\delta[s] \uparrow k) \uparrow o$. We have two cases depending on the type of the strategy $\delta[s] \uparrow k$:

**Case $R^1_e$:** Suppose that $\delta[s] \uparrow k$ is an $R^1_e$-strategy $\alpha$. Let $s^-$ be the previous stage at which $\alpha$ was visited and $o^-$ be the outcome that $\alpha$ had at stage $s^-$. (If $\alpha$ has never been visited then $s^- = 0$ and $o^- = w$). Pick the first case which applies to $\alpha$:

1. If $s_0(\alpha)$ is not defined then set $s_0(\alpha) = s$. End this stage.
2. If $s_1(\alpha)$ is not defined then set $s_1(\alpha) = s$ and $d_\alpha = \omega(s_0(\alpha)) + 1$. Set $C_\alpha = \{(\hat{\alpha}, n) \mid n > s_1(\alpha)\}$ where $\hat{\alpha}$ is the code of $\alpha$ in some fixed computable coding of all nodes on the tree of strategies. End this stage.
3. If $s_1(\alpha)$ is defined, but $X \uparrow s_0(\alpha)$ changed at a stage $t$ such that $s_1(\alpha) < t \leq s$ then restart $\alpha$. End this stage.
4. If $o^- = f$ then let the outcome be $f$.
5. Let $l^\alpha_\alpha$ be the greatest element of the set $\{0\} \cup \{l_\alpha[t] \mid s_1(\alpha) \leq t < s\}$ and $l_\alpha$ be the maximal common initial segment of $\Theta^{\Phi^{A_1}B_1}_e(s)$ and $C_1(s)$. We assume that if $\Theta^{\Phi^{A_1}B_1}_e(n)[s] \downarrow$ then $n < s, \theta(n) < s$ and for all $m < n$ $\Theta^{\Phi^{A_1}B_1}_e(m)[s] \downarrow$. If $l^\alpha_\alpha \geq l_\alpha$ then let the outcome be $w$.
6. If there is an element $n$, such that $\Delta^{\Phi^{A_1}}_e(n) \downarrow G_1(n)$ then pick the least such $n$. Let $\Delta^{\Phi^{A_1}}_e(n) \downarrow$ via the triple $(c, \tau_\theta, \tau_B)$. Enumerate $c$ in $C_1$ and $\tau_B \leq B_1$, by changing the approximation to $B_1$ on numbers $n > s_1(\alpha)$. Let $b_\alpha$ be the least marker $\gamma_1(c)$ that is larger than $|\tau_B|$. Enumerate $b_\alpha$ in $B_1$. For every number $n$ such that $s_1(\alpha) < n < |\tau_B|$ if $n = b_{c'}$ for some chit $c' > s_0(\alpha)$ (belonging to a different strategy), then set $C_1(c') = B_1(b_{c'})$. Enumerate $d_\alpha$ in $G_2$ and $\lambda_2(d_\alpha)$ in $A_2$. Initialize all lower priority strategies and let the outcome be $f$.
7. (7) Scan all $n \leq s$ and perform the following actions for each $n$:

- If $c(n) \uparrow$ or if $G_1(n)[s] \neq G_1(n)[t]$ at some stage $t \geq s_c(n)$ then let $c(n)$ be a fresh number in $C_\alpha$ (one which $\Gamma_1$ has not yet interacted with) and set $g_1(c(n)) = n + 1$ and $s_c(n) = s$. If $\Delta^{\Phi^{A_1}}_e(n) \uparrow$ and $c(n) < l_\alpha$ then let $\tau_\theta = \Phi^{A_1}_e \uparrow \theta(c(n))$ and $\tau_B = B_1 \uparrow \theta(c(n))$. Define $\Delta^{\Phi^{A_1}}_e(n) = G_1(n)$ with use $\tau_\theta$ and say that this computation is defined via the triple $(c(n), \tau_\theta, \tau_B)$.

Once all elements are scanned $\alpha$ ends with outcome $i$.

If $\delta[s] \uparrow k$ is an $R^2_e$-strategy, the instructions are the same as above with $G_1, A_1, B_1, C_1$ swapped with $G_2, A_2, B_2, C_2$.

**Case $Q_e$:** Suppose that $\delta[s] \uparrow k$ is a $Q_e$-strategy $\beta$. Pick the first case which applies to $\beta$:

1. If $s_0(\beta)$ is not defined then set $s_0(\beta) = s$. End this stage.
2. If $s_1(\beta)$ is not defined then set $s_1(\beta) = s$. End this stage.
3. If $X \uparrow s_0(\beta)$ changed at a stage $t$ such that $s_0(\beta) < t \leq s$ then restart $\beta$. End this stage.
(4) If there is a promise \((\tau_X, \tau_G) \in \mathcal{P}_\beta\) which is not currently kept: \(\tau_X \preceq X\) and \(\tau_G \not\preceq (G_1 \oplus G_2)\), then change the value of \((G_1 \oplus G_2)\) to make \(\tau_G \preceq G_1 \oplus G_2\) (enumerate the corresponding \(\Lambda\)-uses in \(A_1\) or \(A_2\)). End this stage.

(5) Let \(l^*_\beta\) be the largest element of the set \(\{0\} \cup \{l_\beta[t] \mid s_1(\beta) \leq t < s\}\) and \(l_\beta\) be the maximal common initial segment of \(\Psi^G_{\beta,1,2}[s]\) and \(Y[s]\). Here we similarly assume that if \(\Psi^G_{\beta,1,2}(x)[s] \downarrow\) then \(x < s\), \(\psi(x) < s\) and for all \(y < x\) \(\Psi^G_{\beta,1,2}(y)[s] \downarrow\). If \(l_\beta > l^*_\beta\) then scan all elements \(x < l_\beta\) and perform the following actions for each:

If \(X(x) \uparrow\) then we define it as \(Y(x)\) with use \(X \uparrow \xi(x)\), where \(\xi(x) > \psi(x)\) is a fresh number. Enumerate in \(P_\beta\) the promise \((X \uparrow \xi(x), (G_1 \oplus G_2) \uparrow \psi(x))\).

End this stage.

(6) If none of the above cases hold, \(\beta\) has outcome \(f\).

We end a stage \(s\) by canceling all strategies of lower priority than \(\delta[s]\) and proceed to stage \(s+1\).

End of construction

6.3. Uniformly low parameters. We have described the construction given a particular pair of approximations \(\{X[s]\}_{s<\omega}\) and \(\{Y[s]\}_{s<\omega}\). Note that the construction can be viewed as a function \(K\) which takes as input the stage \(s\) and outputs \(A_i[s+1], B_i[s+1], C_i[s+1], G_i[s+1]\) and the stage-\(s+1\)-values of all parameters for all strategies. Even if the given sequences of sets \(\{X[s]\}_{s<\omega}\) and \(\{Y[s]\}_{s<\omega}\) do not have the expected properties, the function \(K\) is well defined. We combine all of these constructions together with the additional requirement that the obtained sequence consisting of all possible constructed sets \(B_i\) and \(C_i\) is uniformly low.

For every \(j = (j_x, j_y)\) let \(\{X_j[s]\} = \{j_x\}^{\alpha}[s]\) and \(\{Y_j[s]\} = \{j_y\}^{\alpha}[s]\). We will denote the construction relative to \(\{X_j[s]\} = \{j_x\}^{\alpha}[s]\) and \(\{Y_j[s]\} = \{j_y\}^{\alpha}[s]\) as \(K_j\). We would like to ensure that in addition \(Z = \bigoplus_{s<\omega} B_{j,1} \oplus B_{j,2} \oplus C_{j,1} \oplus C_{j,2}\) is low.

Let \(\{T_i\}\) be a listing of all Turing functionals. For every \(e = \langle i, x \rangle\) we will have the requirement \(L_e\): if there are infinitely many stages \(s\) such that \(T^Z_i(x)[s] \downarrow\) then \(T^Z_i(x) \downarrow\).

Construction:

We will say that \(L_e\) requires attention at stage \(s\) if \(L_e\) is not satisfied at stage \(s\) and \(T^Z_i(x)[s] \downarrow\). Initially all requirements \(L_e\) are not satisfied.

At stage \(s\) check if there is a requirement \(L_e\) with \(e < s\) which requires attention at stage \(s\) and if there is, pick the least one. For all \(j < e\) restart all \(R\)-strategies \(\alpha\) in the tree \(T_j\) which currently have \(s_0(\alpha) \geq e\). For all \(j \geq e\) initialize all strategies in \(T_j\). Announce that \(L_e\) is satisfied.

Run the constructions \(K_j\) for all \(j < s\). Let \(Z[s+1] = \bigoplus_{j \leq s} B_{j,1} \oplus B_{j,2} \oplus C_{j,1} \oplus C_{j,2}[s+1]\). If \(L_e\) is a strategy that was satisfied, \(e = \langle i, x \rangle\) and \(T^Z_i(x) \uparrow\) then announce that \(L_e\) is not satisfied.

End of Construction
6.4. Verification.

Lemma 6.1. For every \( e \), \( \mathcal{L}_e \) is satisfied and requires attention finitely often.

Proof. Fix \( e = (i, x) \). Assume that the statement is true for \( e' < e \). For every \( j < e \) there are finitely many pairs of \((s_0(\alpha_j), \alpha_j)\) such that \( \alpha_j \) is an \( R \)-strategy in \( K_j \) with \( s_0(\alpha_j)[s] < e \) at some stage \( s \). For each such pair the strategy \( \alpha_j \) can change the approximation to \( B_{j,i} \) and \( C_{j,i} \) only once when it performs an attack under Case (6). In order for it to perform a new attack it must be first initialized and will have a different value of the parameter \( s_0(\alpha) \). Suppose that \( s \) is a stage after all of these finitely many strategies have stopped changing the approximation to \( B_{j,i} \) and \( C_{j,i} \) and after all \( \mathcal{L}_{e'} \) where \( e' < e \) have stopped requiring attention. If at all \( t \geq s \) we have \( \Upsilon_Z^e[t] \uparrow \) then \( \mathcal{L}_e \) is satisfied and does not require attention after stage \( s \).

If at stage \( t \geq s \) we see that \( \Upsilon_Z^e[t] \downarrow \) then we restart all strategies with \( s_0(\alpha) \geq e \) in \( T_j \), where \( j < e \) and all strategies in \( T_j \) for \( j > e \). These strategies are activated again after stage \( t \) and can only change the approximation to \( Z \) on numbers larger than \( t \). Thus the computation \( \Upsilon_Z^e(x)[t] \) is preserved at all further stages and \( \mathcal{L}_e \) does not require attention at any further stage. \( \square \)

Let \( K \) be the construction relative to a low \( \Delta^0_2 \) approximation to a set \( X \) and a \( \Delta^0_2 \) approximation to a set \( Y \not\prec_T X \). We prove that the construction \( K \) satisfies all requirements in a series of lemmas. We start with a technical lemma that deals with \( Q \)-strategies and their promises. Once we have established this lemma, we can show that the requirements \( \Lambda_i \) are satisfied and assuming that there is a true path that \( Q \_ \) requirements are satisfied. Next we establish a technical fact about \( R \)-strategies, which in turn will allow us to prove that \( R \)-strategies on the true path satisfy their requirements. We combine these statements to establish the true path and to show that \( \Omega \) is successful. Finally we show that the \( \Gamma_i \) strategies succeed as well.

Lemma 6.2. Suppose that \( \beta \) is a \( Q_\_ \)-strategy, visited at stage \( s \).

1. The promises that \( \beta \) makes are consistent: i.e. if \( \langle \tau_X, \tau_G \rangle, \langle \tau'_X, \tau'_G \rangle \in \mathcal{P}_\beta[s] \) and \( \tau_X \preceq \tau'_X \) then \( \tau_G \) and \( \tau'_G \) are compatible.
2. For every promise \( \langle \tau_X, \tau_G \rangle \in \mathcal{P}_\beta[s] \) we have that \( X[s] \upharpoonright \mathcal{L}_G \leq \tau_G \) and \( G_1 \oplus G_2 \) is compatible with \( \tau_G \).
3. If \( \beta \) changes the value of \( G_1 \oplus G_2 \) on a number \( o \) then \( o \) is the use in some \( \Omega \)-computation and not a threshold.
4. Suppose that \( \langle \tau_X, \tau_G \rangle \in \mathcal{P}_\beta[s^\prime] \cap \mathcal{P}_\beta[s] \), where \( s^\prime < s \). If \( \beta \) keeps this promise stage \( s^\prime \) and \( (X \upharpoonright \tau_X)[s^\prime] = (X \upharpoonright \tau_X)[t] \) for all \( t \in [s^\prime, s] \) then for all \( t \in [s^\prime, s] \) we have \( \tau_G \leq (G_1 \oplus G_2)[t] \).

Proof. (1): We always first ensure that all previous promises are kept before we make new promises. If \( \langle \tau_X, \tau_G \rangle, \langle \tau^\prime_X, \tau^\prime_G \rangle \in \mathcal{P}_\beta \) and \( \tau_X \preceq \tau^\prime_X \) then \( \langle \tau_X, \tau_G \rangle \) was made at stage \( s^0 \) before the promise \( \langle \tau^\prime_X, \tau^\prime_G \rangle \) was made at stage \( s^\prime \). As \( \tau_X \) and \( \tau^\prime_X \) are compatible, it follows that \( \tau_G \leq (G_1 \oplus G_2)[s_2] \); either \( s^0 = s^\prime \) or \( s^0 < s^\prime \) and Case (4) does not apply to \( \beta \) at stage \( s^\prime \) i.e. all promises are kept. As \( \tau^\prime_G \) is also selected as an initial segment of \( (G_1 \oplus G_2)[s^\prime] \), it follows that \( \tau^\prime_G \) is compatible.

(2): Suppose \( \langle \tau_X, \tau_G \rangle \in \mathcal{P}_\beta[s] \) is a promise made at stage \( s^\prime \). Then at stages \( t \in [s^\prime, s] \) the strategy \( \beta \) was not initialized or restarted or else \( \mathcal{P}_\beta[s^\prime] \) would be
cancelled. Hence $X[s'] \uparrow \pi_0(\beta) = X[s] \uparrow \pi_0(\beta)$ and $(G_1 \oplus G_2)[s'] \uparrow \omega(\pi_0(\beta)) = (G_1 \oplus G_2)[s] \uparrow \omega(\pi_0(\beta))$. As $\tau_X$ is selected as an initial segment of $X[s']$ of length larger than $\pi_0(\beta)$ and $\tau_G$ is selected as an initial segment of $(G_1 \oplus G_2)[s']$, the statement follows.

(3): Suppose that at stage $s$ the strategy $\beta$ changes the value of $G_1 \oplus G_2$ at a number $o$. Then $o$ is part of a promise made by $\beta$ at some previous stage $s' < s$ and $\beta$ is not initialized at stages in the interval $[s', s]$. We have that $o < s'$ and there is a least intermediate stage $t$: $s' < t \leq s$ at which the value of $G_1 \oplus G_2$ is changed at $o$ by a strategy $\gamma$. Strategies that change $G_1$ or $G_2$ are $\Omega$, $\mathcal{R}$-strategies or $\mathcal{Q}$-strategies. As $\beta$ is not initialized at stage $t$, $\gamma$ cannot be a higher priority $\mathcal{Q}$- or $\mathcal{R}$-strategy. All lower priority strategies are initialized at stage $s'$ and hence do not change $G_1$ or $G_2$ on elements less than $s'$ at further stages. This follows by construction for $\mathcal{R}$-strategies and by the previous statement in this Lemma for $\mathcal{Q}$-strategies. This leaves $\Omega$ as the only possibility and so $o$ is the use in some $\Omega$-computation.

(4): Finally let $(\tau_X, \tau_G) \in \mathcal{P}_\beta[s'] \cap \mathcal{P}_\beta[s]$, where $s' < s$ and suppose that this promise was made at stage $s'' \leq s'$. At stage $s''$ we know that $\Omega^{\tau_G}$ is an initial segment of $X[s'']$ and by our choice of uses for $\Xi$-computations (as fresh numbers) it follows that $\Omega^{\tau_G} \preceq \tau_X$. At stage $s'$ we keep this promise, hence $\Omega^{\tau_G} \preceq \tau_X \preceq X[s']$. By the previous part of this Lemma it follows that only the $\Omega$-strategy can initiate a series of changes in $G_1 \oplus G_2$ at a promised number at a stage $t \in (s', s]$. It will do so only if $\Omega^{\tau_G} \not\subseteq X[t]$. But since by assumption $(X \uparrow \tau_X) = X[s'] = \tau_X$ it follows that $\tau_G$ remains an initial segment of $(G_1 \oplus G_2)$ at all stages in the interval $[s', s]$. □

Corollary 6.3. $G_i$ is $\Delta^0_2$ and $\Lambda_i = G_i$.

Proof. Fix a number $n$ in $G_i$. The value of $G_i(n)$ changes at most once if $n$ is a threshold for an $\mathcal{R}$ strategy. If $n$ is an $\omega$ marker $o(x)$ then by Lemma 6.2 the number of times that $G_i(n)$ can change is bounded by the number of times $X \uparrow x$ can change. This gives us that $G_i$ is $\Delta^0_2$. Assume inductively that $\Lambda_i(m) \downarrow G_i(m)$ for $m < n$. Every time $G_i(n)$ changes the previous $\Lambda_i$-computation is invalidated forever as the use $\lambda_i(n)$ is enumerated in $A_i$. Let $s$ be a stage such that $G_i \uparrow n$ does not change at further stages. Then $\Lambda_i(n)$ does not change at further stages and so $\Lambda_i(n) \downarrow G_i(n)$. □

Lemma 6.4. Let $\beta$ be a $\mathcal{Q}_e$-strategy which is not initialized after stage $s_0$ and visited infinitely often. Then $\mathcal{Q}_e$ is satisfied and there is a stage $s_3$ after which $\beta$ does not end stages at which it is visited.

Proof. After stage $s_0$ the strategy $\beta$ has a fixed parameter $s_0(\beta)$. As $X$ is $\Delta^0_2$ there will be a least stage $s_1 \geq s_0$ after which $X \uparrow s_0(\beta)$ does not change. At the next $\beta$-true stage after $s_1$ the parameter $s_1(\beta)$ attains its final value. After stage $s_1(\beta)$ Cases (1), (2) and (3) do not apply for $\beta$.

Suppose that there are infinitely many $\beta$-true expansionary stages, i.e. stages at which Case (5) applies to $\beta$. We show that $\Xi^X = Y$, contradicting our choice of $X$ and $Y$. Fix a natural number $x$. Let $s_x > s_1(\beta)$ be the $\beta$-true stage when $\Xi^X(x)$ is first defined. We will show that at every expansionary stage $t \geq s_x$ we have that $\Xi^X(x)[t] \downarrow = Y(x)[t]$. At stage $s_x$ this is true by construction. Suppose that it is true at all $\beta$-true expansionary stage $s_x < t < s$ and consider $s$. If at stage $s$, $\Xi^X(x)[s]$ then we define it to be equal to $Y(x)[s]$. Suppose that $\Xi^X(x)[s] \uparrow$. Then this
computation was defined at a previous expansionary stage \( t \), such that \( s_e \leq t < s \)
using the computation \( \Psi^{G_1,G_2}(x)[t] \). At stage \( t \) the promise \( (X \upharpoonright \xi(x))[t], (G_1 \oplus G_2) \upharpoonright \psi_e(x)[t] \) was made. At stage \( s \) the construction ensures that all promises are kept. As \( (X \upharpoonright \xi(x))[t] \subseteq X[s] \), it follows that \( (G_1 \oplus G_2) \upharpoonright \psi_e(x)[t] \subseteq G_1 \oplus G_2[s] \). Hence:

\[
\Xi^X(x)[s] = \Xi^X(x)[t] = \Psi^{G_1,G_2}(x)[t] = \Psi^{G_1,G_2}(x)[s] = Y(x)[s].
\]

By our choice of low approximation to \( X \) and the fact that \( Y \) is \( \Delta^0_2 \) it follows that
\[
\Xi^X(x) \downarrow = Y(x).
\]

We have shown that there can be only finitely many expansionary stages. Let \( s_e \) be the last \( \beta \)-true expansionary stage. Then for every stage \( t > s_e \), the set of promises does not change: \( P_\beta[t] = P_\beta[s_e] \) and the value of \( l_\beta[t] \) does not change: \( l_\beta[t] = l_\beta[s_e] \). Denote these final values by \( P_\beta \) and \( l_\beta \) respectively. Let \( n \) be the length of the largest \( \tau_X \) in a promise \( \langle \tau_X, \tau_G \rangle \in P_\beta \). Let \( s_X \) be the least stage after \( s_e \) such that \( X \upharpoonright n \) does not change after stage \( s_X \). There are finitely many promises \( \langle \tau_X, \tau_G \rangle \in P_\beta \) with \( \tau_X \leq X \). By part (1) of Lemma 6.2 they are all compatible. Suppose that at stage \( s \geq s_X \) the strategy \( \beta \) keeps a promise \( \langle \tau_X, \tau_G \rangle \) by passing through Case (4). By part (4) of Lemma 6.2 it follows this promise is kept at all further stages. Thus after finitely many passes through Case (4) all of these promises are kept forever. Let \( s_\beta \geq s_e \) be the least stage such that Case (4) does not apply to \( \beta \) at any further stage. Then at stages \( t > s_\beta \), the strategy \( \beta \) always ends in Case (6), it does not end stages prematurely. Furthermore, at all stages \( t > s_\beta \) we have that the length of agreement between \( \Psi^{G_1,G_2}(x)[t] \) and \( Y[t] \) is bounded by \( l_\beta \). It follows that \( \Psi^{G_1,G_2} \neq Y \); the requirement \( Q_e \) is satisfied.

**Lemma 6.5.** Let \( \alpha \) be an \( R^i_\omega \)-strategy visited at stage \( s \) and suppose that \( \alpha \) diagonalizes via Case (6) at stage \( s \). Then \( G_i(n)[t] = G_i(n)[s] \) for all \( n \in [d_\alpha, s] \) and all \( t \geq s \).

**Proof.** Without loss of generality we assume that \( i = 1 \). The strategy \( \alpha \) selects its threshold \( d_\alpha \) at stage \( s_1 < s \) as \( (\omega(s_0(\alpha)) + 1)[s_1] \). The strategy is not initialized until at stage \( s \) it enumerates \( d_\alpha \) in \( G_2 \). First we show that \( d_\alpha \) remains in \( G_2 \) at all further stages. By construction when an \( R \)-strategy selects the value of its parameter \( s_0 \) it ends the stage. Thus \( \alpha \) is the unique strategy whose threshold is selected relative to the use of \( s_0 \). Note that \( d_\alpha \) is not the use in any \( \Omega \)-computation. Thus by Proposition 6.2 no \( Q \)-strategy can extract \( d_\alpha \) from \( G_2 \) at any further stage.

Let \( n \in [d_\alpha, s] \) be a number and towards a contradiction assume that \( G_1(n)[t] \neq G_1(n)[s] \), where \( t > s \). The possible roles of \( n \) are - a threshold for an \( R \)-strategy \( \alpha' \), an \( \Omega \)-use and a member of a promise in a \( Q \)-strategy \( \beta \). We rule these out in turn: If \( n \) is a threshold for \( \alpha' \) then it is selected at a stage in the interval \( [s_1, s] \).

As \( \alpha \) is not initialized until stage \( s \), \( \alpha' \) cannot be of higher priority. If \( \alpha' \) is of lower priority than \( \alpha \) then it is initialized at stage \( s \), its threshold is canceled and hence \( \alpha' \) cannot change the value of \( G_1 \) at \( d_\alpha \). If \( n \) is the \( \omega \)-use of some computation in \( \Omega \) then this computation is defined in the interval \( [s_1, s] \) and hence must involve the fact that \( G_2(d_\alpha) = 0 \) at these stages. After stage \( s \) this computation is invalid as \( d_\alpha \in G_2 \) and hence \( \Omega \) will never need to change the value of \( G_1(n) \). Finally suppose that \( \beta \) changes the approximation to \( G_1(n) \) after stage \( s \), due to a promise \( \langle \tau_X, \tau_G \rangle \) with \( \tau_G(2n) \neq G_1(n)[s] \). Assume that \( \beta \) is the first \( Q \)-strategy to do so. Consider the stage \( t_0 \) when \( \beta \) made this promise. It must be that \( t_0 > s_0(\alpha)[s_1] \) as \( n \geq d_\alpha \).

At this stage \( \beta \) observed that \( G_1(n) \) has a different value, i.e. \( G_1(n)[t_0] \neq G_1(n)[s] \).
and this is not due to \( \beta \)'s action. As we have already ruled out any other reason for such a change at stages greater than \( s \), it follows that \( t_0 \leq s \). Whenever \( \beta \) makes a new promise, it ends the stage thereby initializing all lower priority strategies, thus \( \beta \) cannot be of higher priority than \( \alpha \). But then \( \beta \) is initialized at stage \( s \) and its list of promises is emptied. \( \square \)

**Lemma 6.6.** Let \( \alpha \) be an \( R^i_e \)-strategy which is not initialized after stage \( s_0 \) and visited infinitely often. Then \( R^i_e \) is satisfied and there is a stage \( s_\alpha \) after which \( \alpha \) does not end stages at which it is visited.

*Proof.* Again for concreteness we will assume that \( i = 1 \). After stage \( s_0 \) the parameter \( s_0(\alpha) \) does not change and higher priority \( R \)- and \( Q \)-strategies do not make any further changes. Consider any of the global parameters: \( A_i, C_i, B_i, \) and \( G_i \). As \( X \) is \( \Delta^2_0 \) and by Lemma 6.1 there will be a least stage \( s_1 \geq s_0 \) after which \( X \mid s_0(\alpha) \) does not change and strategies \( L_i \) where \( i < s_0(\alpha) \) do not require attention. At the next \( \alpha \)-true stage after \( s_1 \) the parameters \( s_1(\alpha) \), \( d_\alpha \) and \( C_\alpha \) attain their final values.

The final value of \( d_\alpha \) is \( \omega(s_0(\alpha))[s_1] + 1 \). As \( X \mid s_0 \) does not change at any further stage and all lower priority strategies are initialized at stage \( s_1(\alpha) \) it follows that \( G_1 \mid d_\alpha \) and hence \( A_1 \mid d_\alpha \) do not change at any further stage. Similarly \( B_1 \mid s_1(\alpha) \) and \( C_1 \mid s_1(\alpha) \) do not change at any further stage. After stage \( s_1(\alpha) \) Cases (1), (2) and (3) do not apply for \( \alpha \).

Suppose that Case (4) applies for \( \alpha \) at a least stage \( s_4 \). Then at all further stages \( t \geq s_4 \) \( \alpha \) ends with outcome \( f \) and does not initialize lower priority strategies. Consider the \( \alpha \)-true stage before \( s_4 \), call it \( s \). At stage \( s \) Case (6) applies to \( \alpha \). There is an element \( n \) such that \( \Delta^\Phi_{\Phi^A \cdot B_1}(n) \downarrow \) via the triple \( (c, \tau_\Phi, \tau_B) \). This computation was observed at a previous stage \( s' \), such that \( s_0(\alpha) < s' < s \) and so \( B_1[s] \mid s_0(\alpha) \leq \tau_B \).

At stage \( s \) \( \alpha \) changes the value of \( B_1 \) on numbers larger than \( s_0(\alpha) \) to ensure that \( \tau_B \leq B_1[s] \). Thus \( \Theta^\Phi_{\Phi^A \cdot B_1}(c) \downarrow = 0 \). By Lemma 6.5 \( G_1 \mid s \) and hence \( A_1 \mid s \) is preserved at all further stages. At stage \( s \) we initialize all lower priority strategies, so \( B_1 \mid s \) is also preserved. Thus \( \Theta^\Phi_{\Phi^A \cdot B_1}(c)[s] \) is preserved at all further stages and \( \Theta^\Phi_{\Phi^A \cdot B_1} \neq C_1 \).

If Case (4) never applies for \( \alpha \) after stage \( s_1(\alpha) \) then neither does Case (6).

Hence \( \alpha \) does not end prematurely any true stage \( t > s_1(\alpha) \).

Suppose that there is a stage \( s_5 \) such that at all \( \alpha \)-true stages \( t > s_5 \) Case (5) applies for \( \alpha \). It follows that the value of \( l_\alpha \) does not change after stage \( s_5 \) and hence at all \( t > s_5 \) there is a number \( n < l_\alpha \) such that \( (\Theta^\Phi_{\Phi^A \cdot B_1}(n))[t] \uparrow \) or \( (\Theta^\Phi_{\Phi^A \cdot B_1}(n))[t] \downarrow = C_1(n)[t] \). Hence \( \Theta^\Phi_{\Phi^A \cdot B_1} \) is not total or not equal to \( C_1 \) and \( R^i_e \) is satisfied.

Finally suppose that at infinitely many \( \alpha \)-true stages Case (7) applies. We will show that if \( \Theta^\Phi_{\Phi^A \cdot B_1} \) is total then \( \Delta^\Phi_{\Phi^A \cdot B_1} \) is total and equal to \( G_1 \). Fix a natural number \( n \). At every \( \alpha \)-true expansionary stage \( s \) we have that if \( \Delta^\Phi_{\Phi^A \cdot B_1}(n)[s] \downarrow \) then \( \Delta^\Phi_{\Phi^A \cdot B_1}(n)[s] = G_1(n) \) or else Case (6) would apply. As \( G_1 \) is \( \Delta^2_0 \) we can fix a stage \( s_n \) such that \( G_1 \mid n \) does not change at any further stage. At the least \( \alpha \)-true expansionary stage after \( s_n \) the final value of the chit \( c(n) \) will be defined. As \( \Theta^\Phi_{\Phi^A \cdot B_1}(c(n)) \downarrow \) there will be a stage \( s_\epsilon \geq s_n \) such that the computation \( \Theta^\Phi_{\Phi^A \cdot B_1}(c(n)) \) does not change. If at any stage \( t > s_\epsilon \) \( \Delta^\Phi_{\Phi^A \cdot B_1}(n) \) becomes undefined.
then it will be redefined using the correct computation for \( c(n) \) and hence will not become undefined at further stages. Thus \( \Delta^{\Phi_{\alpha_1}}_\alpha = G_1 \). □

From Lemma 6.1, Lemma 6.4 and Lemma 6.6 we get immediately the True Path Lemma.

**Corollary 6.7.** There is an infinite path \( f \) in the tree such that for every \( n \), \( f \upharpoonright n \) is visited infinitely often and initialized only finitely often. All \( \mathcal{R} \) and \( \mathcal{Q} \) requirements are satisfied.

**Lemma 6.8.** \( \Omega^{G_1,G_2}(n) = X(n) \).

**Proof.** At every stage \( s \) the \( \Omega \) strategy ensures that for every \( n < s \), \( \Omega^{G_1,G_2}(n)[s] \downarrow = X(n)[s] \). We need to show that \( \Omega \) is total. Fix a natural number \( n \). Consider \( f \upharpoonright n \) and let \( s_0 = s_0(f \upharpoonright n) \) and \( s_1 = s_1(f \upharpoonright n) \) be the final values of these parameters. It follows from Lemma 6.4 and Lemma 6.6 that \( (G_1 \oplus G_2)[s_1] \upharpoonright \omega(s_0) \) does not change after stage \( s_1 \). Hence \( \Omega^{G_1,G_2}(n) \downarrow \). □

**Lemma 6.9.** \( \Gamma_i^{B_i,G_i} = C_i \)

**Proof.** Let \( c \) be a natural number. Suppose that \( c \) is enumerated in \( C_i \) at stage \( s \) by a strategy \( \alpha \). Then \( \alpha \) is a chit for a number \( n \) and \( \alpha \) attacks in relation to a computation defined using \( (c, \tau_B, \tau_B) \). At stage \( s \) the strategy \( \alpha \) defines the value of \( b_c \). By the way that chits are assigned to numbers it follows that every \( \Gamma_i \)-computation defined for \( c \) until stage \( s \) either uses the fact that \( G_i(n) = 1 - G_i(n)[s] \) or uses the fact that \( B_i(b_c) = 0 \). By Lemma 6.5 the value of \( G_i(n) \) is the same at all stages \( t \geq s \) hence computations of the first kind are never valid again. At stage \( s \) \( \alpha \) initializes all lower priority strategies and so they can never again change the value of \( B_i(b_c) \). From this point on we maintain that \( B_i(b_c) = C_i(c) \).

There are finitely many strategies \( \alpha' \), the ones of higher priority than \( \alpha \), that can change the value of \( B_i(b_c) \) at a stage \( t > s \) and each can do this only once when it diagonalizes under Case (6), given that it is not initialized in the interval \([s,t] \). It follows that when such a strategy attacks the value of its parameter \( s_1(\alpha') \) is smaller than \( c \). So if \( \alpha' \) changes the value of \( B_i(b_c) \) then it also ensures \( C_i(c) = B_i(b_c) \). After all these finitely many strategies are done \( B_i(b_c) \) and \( C_i(c) \) stop changing.

It follows that \( B_i \) and \( C_i \) are \( \Delta^0 \) and at all stages \( t > s \) if \( \Gamma_i^{B_i,G_i}(c)[t] \downarrow \) then \( \Gamma_i^{B_i,G_i}(c)[t] = C_i(c)[t] = B_i(b_c)[t] \).

Suppose inductively that \( \Gamma_i^{B_i,C_i}(d) \downarrow = C_i(d) \) for all \( d < c \). Suppose that \( s_1 \) is a stage such that:

1. All computations for \( d < c \) have settled.
2. \( G_i \upharpoonright g_i(n) \) does not change at further stages.
3. \( b_c \uparrow \) or \( B_i \upharpoonright b_c \) does not change at further stages.
4. \( C_i \upharpoonright c \) does not change at further stages.

Then at stages \( t > s \) the value of \( \gamma_i(c) \) does not change. Let \( s_2 \geq s_1 \) be such that \( B_i \upharpoonright \gamma_i(c) \) does not change at further stages. If \( \Gamma_i^{B_i,G_i}(c)[s_2] \downarrow \) and \( b_c \uparrow \) then at all stages until \( s_2 \) we have \( C_i(c) = 0 \), so every computation defined until \( s_2 \) is of the right value of \( C_i(n) \). If \( b_c \downarrow \) then by the discussion above \( \Gamma_i^{B_i,G_i}(c)[s_2] = B_i(b_c)[s] = C_i(c)[s_2] \). If \( \Gamma_i^{B_i,G_i}(c)[s_2] \uparrow \) then we define it for the last time correctly and it remains valid at all further stages. □
References


(Slaman) Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720-3840, USA
E-mail address, Slaman: slaman@math.berkeley.edu

(Soskova) Faculty of Mathematics and Informatics, Sofia University, 5 James Bourchier Blvd., 1164 Sofia, Bulgaria
E-mail address, Soskova: msoskova@fmi.uni-sofia.bg