The arithmetical hierarhy in the setting of ω_1

Jacob Carson, Jesse Johnson, Julia F. Knight, Karen Lange, Charles McCoy CSC, John Wallbaum

University of Notre Dame

Abstract

We continue work from [1] on computable structure theory in the setting of ω_1, where the countable ordinals play the role of natural numbers, and countable sets play the role of finite sets. In the present paper, we define the arithmetical hierarchy—through all countable levels. We do this in two different ways. We then give analogues of the well-known results from [2] and [4] on relatively intrinsically arithmetical relations. To do this, we define computable Σ_α and computable Π_α formulas, for countable ordinals α. We do this also in two different ways, corresponding to our two definitions of the arithmetical hierarchy. We prove, for both approaches, that a relation R on a computable structure $\mathcal{A}$ is relatively intrinsically Σ^0_α if and only if it is defined in $\mathcal{A}$ by a computable Σ_α formula.

1 Introduction

We consider computability in the setting where countable ordinals play the role of natural numbers, and countable sets play the role of finite sets. It will be beneficial for us to assume that every subset of ω_1 is amenable for L_{ω_1}. So we assume $V = L$, or at least $P(\omega) \subseteq L$. The reason for this assumption is that, since L_{ω_1} is the domain for our model of computation, it will be important for our purposes that for all $A \subseteq L_{\omega_1}$, L_{ω_1} will be closed under the function $x \mapsto A \cap x$. In the remainder of the introduction, we review some basic definitions and results from [1]. In Section 2, we define the arithmetical hierarchy in two different ways, one resembling the standard definition of the hyperarithmetical hierarchy, and the other resembling the definition of the effective Borel hierarchy. In Section 3, we define computable infinitary formulas, allowing countable tuples of variables. We do this in two different ways, corresponding to the different definitions of the arithmetical hierarchy. In Section 4, we prove the main result, saying that a relation R on a computable structure $\mathcal{A}$ is relatively intrinsically Σ_α if and only if it is defined in $\mathcal{A}$ by a computable Σ_α formula. There are two versions of the result, one for each set of definitions. This main result is an analogue to a result in the standard computability setting in [2] and [4]. Finally, in Section 5, we explain why the second set of definitions seems preferable.[1]

1.1 Basic definitions

Below, we first say what it means for a set or relation to be computably enumerable. We then define the computable sets and relations, and the computable functions.

Definition 1.

- *A relation $R \subseteq (L_{\omega_1})^n$ is* computably enumerable, *or* c.e., *if it is defined in L_{ω_1} by a Σ_1-formula $\varphi(\bar{c}, x)$, with finitely many parameters—the formula is finitary, with only existential and bounded quantifiers.*

- *A relation $R \subseteq (L_{\omega_1})^n$ is* computable *if R and the complementary relation $\neg R$ are both computably enumerable.*

- *A (partial) function $f : (L_{\omega_1})^n \to L_{\omega_1}$ is* computable *if its graph is c.e.*

When we work with these definitions, we soon see that computations involve countable ordinal steps. Computable functions are defined by recursion on ordinals—the Σ_1 definition for a function f says that there is a sequence of steps leading to the value of f at a given ordinal α. Thus, it might be appropriate to use the term "recursive" instead of "computable" in this setting.

1.2 Ordinals and sets

Results of Gödel give us a computable $1-1$ function g from the countable ordinals onto L_{ω_1} such that the relation $g(\alpha) \in g(\beta)$ is computable. The function g gives us ordinal codes for sets—α is the code for $g(\alpha)$. There is also a computable function ℓ taking α to the code for L_α. From this, we see that computing in ω_1 is essentially the same as computing in L_{ω_1}. For a more explicit argument, see [1].

1.3 Arity of relations and functions

We have been thinking of relations and functions of finite arity. However, we may allow relations and functions of arity α, for any $\alpha < \omega_1$. We extend the definition as follows.

Definition 2. *Suppose R is a relation of arity $\alpha < \omega_1$.*

- *R is c.e. if $\{\beta : g(\beta) \in R\}$ is Σ_1-definable; i.e., the set of ordinal codes for sequences in R is c.e.*

- *R is computable if it is both c.e. and co-c.e.*

- *A function $f : (L_{\omega_1})^\alpha \to L_{\omega_1}$ is computable if its graph $\{\overline{a}, f(\overline{a}) : \overline{a} \in dom f\}$ is a c.e. relation.*

[1]We are grateful to Joe Mileti for helpful discussions of this.

1.4 Indices for c.e. sets

- We have a c.e. set C of codes for pairs $(\varphi, \overline{c})$, each representing Σ_1 definitions, where $\varphi(\overline{u}, \overline{x})$ is a Σ_1-formula and $\overline{c}$ is a tuple of parameters appropriate for $\overline{u}$. Note that $\overline{u}$ and $\overline{x}$ can be countable tuples.
- We have a computable function h mapping ω_1 onto C.

Definition 3. *α is a* c.e. index for X *if $h(\alpha)$ is the code for a pair $(\varphi, \overline{c})$, where $\varphi(\overline{c}, x)$ is a Σ_1 definition of X in $(L_{\omega_1}, \in)$.*

Notation: We write W_α for the c.e. set with index α.

1.5 Stage at which an element enters a c.e. set

Suppose W_α is determined by the pair $(\varphi, \overline{c})$; i.e., $\varphi(\overline{c}, x)$ is a Σ_1 definition.

Definition 4. *We say that x* is in W_α at stage β, *and we write $x \in W_{\alpha,\beta}$, if L_β contains x, the parameters $\overline{c}$, and witnesses making the formula $\varphi(\overline{c}, x)$ true.*

Remark. The relation $x \in W_{\alpha,\beta}$ is computable. The relation is Σ_1 definable, and so is the complementary relation—we use L_β as a bound on the quantifiers.

1.6 Universal enumeration of c.e. sets, and K

Proposition 1.1. *There is a c.e. set $U \subseteq \omega_1 \times \omega_1$ consisting of the pairs (α, β) such that $\beta \in W_\alpha$.*

We define the halting set, just as in the standard setting.

Definition 5 (Halting set)**.** *The* halting set *is $K = \{\alpha : \alpha \in W_\alpha\}$.*

Just as in the standard setting, K is c.e. and not computable.

1.7 Relative computability

We define relative computability in a natural way.

Definition 6. *Let $X \subseteq \omega_1$ and let R be a relation.*

- *R is* c.e. relative to X *if it is Σ_1-definable in $(L_{\omega_1}, \in, X)$.*
- *R is* computable relative to X *if R and the complementary relation $\neg R$ are both c.e. relative to X.*
- *A (partial or total) function is* computable relative to X *if the graph is c.e. relative to X.*

Definition 7. *A* c.e. index *for a relation R relative to X is an ordinal α such that $h(\alpha) = (\varphi, \overline{c})$, where φ is a Σ_1 formula (in the language with $\in$ and a predicate symbol for X), and $\varphi(\overline{c}, x)$ defines R in $(L_{\omega_1}, \in, X)$.*

Notation: We write W_α^X for the c.e. set with index α relative to X.

Proposition 1.2. *There is a c.e. set U of the codes for triples (σ, α, β) such that $\sigma \in 2^\rho$ (for some countable ordinal ρ), and for all X with characteristic function extending σ, $\beta \in W_\alpha^X$. A proof appears in [1].*

1.8 Jumps

We define the *jump* of a set $X \subseteq \omega_1$ so that it is clearly complete.

Definition 8. *The* jump *of X is $X' = \oplus_{\alpha<\omega_1} W_\alpha^X = \{(\alpha, x) : x \in W_\alpha^X\}$.*

As in the standard setting, X' is c.e. relative to X, and any set c.e. relative to X is m-reducible to X'.

We iterate the jump function through countable levels as follows.

- $X^{(0)} = X$,
- $X^{(\alpha+1)} = (X^{(\alpha)})'$,
- for limit α, $X^{(\alpha)}$ is the set of codes for pairs (β, x) such that $\beta < \alpha$ and $x \in X^{(\beta)}$.

Notation. For convenience, we write Δ_n^0 for $\emptyset^{(n-1)}$ for $1 \leq n < \omega$, and we write Δ_α^0 for $\emptyset^{(\alpha)}$ for countable ordinals $\alpha \geq \omega$.

2 Arithmetical hierarchy—two definitions

In this section, we give two different definitions of the arithmetical hierarchy.

2.1 First approach

In our first definition, we imitate the approach used in defining the effective Borel hierarchy [7].

Definition 9. *Let R be a relation on ω_1.*

- *R is Σ_1^0 if it is c.e.; R is Π_1^0 if the complementary relation, $\neg R$, is c.e.*
- *For countable $\alpha > 1$, R is Σ_α^0 if it is a c.e. union of relations, each of which is Π_β^0 for some $\beta < \alpha$. That is, $R = \bigcup_{c.e.} R_i$, where for each i, there is some $\beta_i < \alpha$ such that R_i is $\Pi_{\beta_i}^0$; R is Π_α^0 if it is a c.e. intersection of relations, each of which is Σ_β^0 for some $\beta < \alpha$.*
- *R is Δ_α^0 if it is both Σ_α^0 and Π_α^0.*

Remarks. In general, when we form a Σ^0_α relation, we are taking the union of an *uncountable* c.e. family of lower complexity relations. A relation is Π^0_α just in case the complementary relation, $\neg R$, is Σ^0_α.

We may assign indices to the Σ^0_α and Π^0_α sets in a natural way, for $\alpha \geq 1$. The indices have the form (Σ, α, γ) and (Π, α, γ). The first two components indicate that the set is Σ^0_α, or Π^0_α. For $\alpha = 1$, the set with index $(\Sigma, 1, \gamma)$ is the c.e. set with index γ, and the set with index $(\Pi, 1, \gamma)$ is the complement. Suppose $\alpha > 1$. Then set with index (Σ, α, γ) is the union of the sets with indices in W_γ of the form (Π, β, δ) for $1 \leq \beta < \alpha$. The set with index (Π, α, γ) is the intersection of the c.e.
sets with indices in W_γ of the form (Σ, β, δ).

The definition relativizes in a natural way. A relation R is $\Sigma^0_1(X)$ if it is c.e. relative to X, it is $\Pi^0_1(X)$ if the complementary relation $\neg R$ is c.e. relative to X. For $\alpha > 1$, R is $\Sigma^0_\alpha(X)$ if it is a c.e. union of relations each of which is $\Pi^0_\beta(X)$ for some $\beta < \alpha$, etc.

2.2 Second approach

In our second definition for the arithmetical hierarchy, we follow the approach used in defining the hyperarithmetical hierarchy in the usual setting.

Definition 10. *Let R be a relation on ω_1.*

- *R is Σ^0_1 if it is c.e.; R is Π^0_1 if $\neg R$ is c.e.*
- *For countable $\alpha > 1$, R is Σ^0_α if it is c.e. relative to Δ^0_α; R is Π^0_α if $\neg R$ is Σ^0_α.*
- *R is Δ^0_α if it is both Σ^0_α and Π^0_α.*

Again we may assign indices to the Σ^0_α and Π^0_α sets in a natural way.The form is the same as above. For $\alpha \geq 1$, the set with index (Σ, α, γ) is $W_\gamma^{\Delta^0_\alpha}$, and the set with index (Π, α, γ) is the complementary set.

The second definition also relativizes in a natural way. A relation R is $\Sigma^0_1(X)$ if it is c.e. relative to X, and $Pi^0_1(X)$ if the complement is c.e. relative to X. For $\alpha > 1$, R is $\Sigma^0_\alpha(X)$ if it is c.e. relative to the natural oracle that is complete $\Delta^0_\alpha(X)$. We write $\Delta^0_\alpha(X)$ for this oracle. Here is the precise definition.

Definition 11 (Oracle $\Delta^0_\alpha(X)$)**.** *For $n = 1$, $\Delta^0_1(X)$ for X, and for finite $n > 1$, we write $\Delta^0_n(X)$ for $X^{(n-1)}$. For $\alpha \geq \omega$, we write $\Delta^0_\alpha(X)$ for X^α.*

2.3 Comparing the two definitions

The two definitions agree at finite levels, but they disagree at level ω and beyond. Note that under the first definition, each element of a Σ^0_α set must be in one of the lower Π^0_β sets whose union we are taking. This means that for each element, membership in the Σ^0_α set uses information from a single lower level. Under the second definition, as in the standard setting, putting a single element into a Σ^0_α set may use information from all lower levels. In particular,

- Under the first definition, putting an element into a Σ^0_ω set uses information from a single finite level.
- Under the second definition, putting an element into a Σ^0_ω set may use information at all finite levels.

Proposition 2.1. *There is a set that is Δ^0_ω under the second definition but not Σ^0_ω under the first definition.*

Proof. Each set that is Σ^0_ω under the first definition has an index of the form (Σ, ω, α)—the set is equal to the union of the sets with indices in W_α of the form (Π, n, β), for $n \in \omega$. We define a set S, Δ^0_ω under the second definition, such that $\alpha \in S$ iff α is not in the set with first-definition index (Σ, ω, α). Thus S is defined to to diagonalize against the sets that are Σ^0_ω under the first definition. For each α and n, let $S_{(\alpha,n)}$ be the union of the sets with indices in $(\Pi, k, \beta) \in W_\alpha$, for $k < n$ for this fixed n. Note that each $S_{(\alpha,n)}$ is Σ^0_n. The union of these sets over all n will be the Σ^0_ω set with index (Σ, ω, α). For each countable α, we can determine, computably relative to Δ^0_ω, whether $(\forall n \in \omega)\,(\alpha \notin S_{(\alpha,n)})$. Answering countably many questions in the current setting is like answering finitely many questions in the standard setting. We define S such that $\alpha \in S$ iff $\alpha \notin S_{(\alpha,n)}$ for all n. This S is Δ^0_ω under the second definition, but it is not even Σ^0_ω under the first definition. □

3 Computable infinitary formulas

In this section, we define the computable infinitary Σ_α and Π_α formulas for countable ordinals α. We do this in two different ways, corresponding to our two different definitions of the arithmetical hierarchy. For information on computable infinitary formulas in the standard setting, see [3]. We consider both predicate and propositional formulas. The formulas will be in "normal form" in which the negations occur only next to atomic formulas.

3.1 First approach

Our first definition of the computable infinitary formulas corresponds to our first definition of the arithmetical hierarchy. We consider both predicate and propositional languages, since we shall use both for the results in Section 4.

Predicate formulas

Let L be a predicate language. For simplicity, we suppose that the symbols are as usual, with *finite* arity. We suppose that the set of symbols in L is computable, and that determining the type and arity of symbols in L is computable. Recall that in $L_{\omega_1,\omega}$, we allow countable disjunctions and conjunctions, but only finite nesting of quantifiers. For more imformation on $L_{\omega_1,\omega}$, see [3]. In the current setting, we want to allow possibly uncountable but c.e. disjunctions and conjunctions. We also allow some countable nesting of quantifiers. The formulas from $L_{\omega_1\omega}$ that we use are in "normal form", with negations occurring only next to atomic formulas. Here is an informal description of the computable infinitary formulas.

Definition 12 (Computable infinitary predicate formulas)**.**

- $\varphi(\overline{x})$ *is* computable Σ_0 *and* computable Π_0 *if it is a quantifier-free formula of* $L_{\omega_1,\omega}$ *(in normal form).*
- *For* $\alpha > 0$,
  - *(i)* $\varphi(\overline{x})$ *is* computable Σ_α *if it is a c.e. disjunction of formulas* $(\exists\overline{u})\,\psi(\overline{u},\overline{x})$, *where* ψ *is computable* Π_β *for some* $\beta < \alpha$. *Here, by a c.e. disjunction we mean a disjunction of formulas whose codes are given by some c.e. set.*
  - *(ii)* $\varphi(\overline{x})$ *is* computable Π_α *if it is a c.e. conjunction of formulas* $(\forall\overline{u})\,\psi(\overline{u},\overline{x})$, *where* ψ *is computable* Σ_β *for some* $\beta < \alpha$. *Likewise, by a c.e. conjunction we mean a conjunction of formulas whose codes are given by some c.e. set.*

To make the definition above precise, we need indices. A quantifier-free formula $\varphi(\overline{x})$ of $L_{\omega_1\omega}$ has a natural representation as a set theoretic object (a parsing tree). This will be an element of L_{ω_1}, so it has a code β. As computable Σ_0 and Π_0 indices for the formula, we take $(\Sigma, 0, \overline{x}, \beta)$ and $(\Pi, 0, \overline{x}, \beta)$. For $\alpha > 0$, the computable Σ_α formula with index $(\Sigma, \alpha, \overline{x}, \gamma)$ represents the disjunction over formulas $(\exists\overline{u})\,\psi(\overline{x},\overline{u})$ such that $\psi(\overline{x},\overline{u})$ has an index in W_γ of the form $(\Pi, \beta, \overline{x}, \overline{u}, \delta)$, for $\beta < \alpha$. The computable Π_α formula with index $(\Pi, \alpha, \overline{x}, \gamma)$ represents the disjunction over formulas $(\forall\overline{u})\,\psi(\overline{x},\overline{u})$ such that $\psi(\overline{x},\overline{u})$ has an index in W_γ of the form $(\Sigma, \beta, \overline{x}, \overline{u}, \delta)$, for $\beta < \alpha$.

Given a formula φ, we shall write $neg(\varphi)$ for the "dual" formula in which

- disjunctions and conjunctions are interchanged,
- existential and universal quantifiers are interchanged,
- atomic formulas and negations of atomic formulas are interchanged.

The formula $neg(\varphi)$ is logically equivalent to the negation. It is not difficult to see that if φ is computable Σ_α, then $neg(\varphi)$ is computable Π_α, and vice versa.

In what follows, we consider structures $\mathcal{A}$ with universe a subset of ω_1 (or L_{ω_1}). As in the standard setting, when we want to think about complexity, we identify $\mathcal{A}$ with its atomic diagram, or with the set of codes for these sentences.

In the result below, the definitions are as in the first approach.

Proposition 3.1. *Let $\mathcal{A}$ be an L-structure, and let $\varphi(\overline{x})$ be a computable infinitary L-formula. If $\varphi(\overline{x})$ is computable Σ_α (or computable Π_α), for $\alpha \geq 1$, then the relation defined by $\varphi(\overline{x})$ in $\mathcal{A}$ is Σ^0_α (or Π^0_α) relative to $\mathcal{A}$, uniformly in φ.*

Proof. For the proof in the standard setting, see [3], pg. 110. The proof in our setting is similar. If $\varphi(\overline{x})$ is a quantifier-free formula of $L_{\omega_1\omega}$, then to decide, for a countable tuple $\overline{a}$ in $\mathcal{A}$ whether $\mathcal{A} \models \varphi(\overline{a})$, we find a countable substructure $\mathcal{A}'$ containing $\overline{a}$ and check whether $\mathcal{A}' \models \varphi(\overline{a})$. The relation defined by $\varphi(\overline{x})$ is computable relative to $\mathcal{A}$.

Suppose $\varphi(\overline{x})$ is computable Σ_1, the disjunction of a c.e. set of formulas $(\exists \overline{u}_i)\, \psi_i(\overline{x}, \overline{u}_i)$, where ψ_i is a quantifier-free formula of $L_{\omega_1\omega}$. For $\overline{a}$ in $\mathcal{A}$, we have $\mathcal{A} \models \varphi(\overline{a})$ iff there exist i and a tuple $\overline{b}_i$ such that $\psi_i(\overline{x}, \overline{b}_i)$. The relation defined by $\varphi(\overline{x})$ is Σ_1-definable in $(L_{\omega_1}, \mathcal{A})$, so it is c.e. relative to $\mathcal{A}$. If $\varphi(\overline{x})$ is computable Π_1, the conjunction of a c.e. set of formulas $(\forall \overline{u}_i)\, \psi_i(\overline{x}, \overline{u}_i)$, we can see that the relation defined by $\varphi(\overline{x})$ is the complement of a relation c.e. relative to $\mathcal{A}$.

We proceed by induction on $\alpha > 1$. Let $\varphi(\overline{x})$ be computable Σ_α, the disjunction of a c.e. set of formulas $(\exists \overline{u})\, \psi_i(\overline{u}_i, \overline{x})$, where each ψ_i is computable Π_β for some $1 \leq \beta < \alpha$. The formula $(\exists \overline{u})\, \psi_i(\overline{u}_i, \overline{x})$ is computable Σ_{β_i} for some $\beta_i < \alpha$. By the Induction Hypothesis, the relation defined by $\psi_i(\overline{u}_i, \overline{x})$ is $\Pi^0_{\beta_i}$ relative to $\mathcal{A}$, uniformly. Then the relation defined by $(\exists \overline{u}_i)\, \psi_i(\overline{x}, \overline{u}_i)$ is the union over the possible i and $\overline{b}_i$ of the relations $\overline{b} \in \mathcal{A} \rightarrow \mathcal{A} \models \psi_i(\overline{x}, \overline{b}_i)$. This is Σ^0_α.

If $\varphi(\overline{x})$ is computable Π_α, we consider $neg(\varphi)$, which has the dual form and is logically equivalent to the negation of φ. Now, $neg(\varphi)$ is computable Σ_α, so the relation that it defines is Σ^0_α relative to $\mathcal{A}$. The relation defined by φ itself is the complement of the relation defined by $neg(\varphi)$, so it is Π^0_α.

□

Propositional formulas

Recall that in P_{ω_1}, the formulas are built up from the formulas p and $\neg p$, for $p \in P$, using countable conjunctions and disjunctions. For more imformation on P_{ω_1}, see [3]. We may consider formulas from P_{ω_1} in "normal" form, with negations occurring only next to propositional variables. Let P be a propositional language with a computable set of propositional variables, not necessarily countable.

Definition 13 (Computable infinitary propositional formulas)**.**

- *φ is* computable Σ_0 *and* computable Π_0 *if it is a formula of P_{ω_1}.*
- *For $\alpha > 0$,*

  (i) φ is computable Σ_α *if it is a c.e. disjunction of formulas each of which is computable Π_β for some $\beta < \alpha$.*

  (ii) φ is computable Π_α *if it is a c.e. conjunction of formulas each of which is computable Σ_β for some $\beta < \alpha$.*

Again we need indices to make this precise. A formula φ of P_{ω_1} has a natural representation as a set theoretic object (a parsing tree). This will be an element of L_{ω_1}, so it has a code β. The formula φ involves only countably many propositional variables. For $\alpha > 0$, the computable Σ_α formula with index (Σ, α, γ) represents the disjunction over the formulas ψ with an index in W_γ of the form (Π, β, δ), for $\beta < \alpha$. The computable Π_α formula with index (Π, α, γ) represents the disjunction over the formulas ψ with an index in W_γ of the form (Σ, β, δ), for $\beta < \alpha$.

Again, our formulas are in "normal" form, with negations only next to the propositional variables. For each formula φ, the formula $neg(\varphi)$ is the dual formula that is logically equivalent to the negation of φ.

3.2 Second approach

Our second definition of the computable infinitary formulas corresponds to our second definition of the arithmetical hierarchy. Again we consider both predicate and propositional languages.

Predicate formulas

Let L be a computable predicate language, as above.

Definition 14 (Computable infinitary predicate formulas)**.**

- *$\varphi(\overline{x})$ is* computable Σ_0 *and* computable Π_0 *if it is a quantifier-free formula of $L_{\omega_1\omega}$.*
- *For $\alpha > 0$,*

  (i) $\varphi(\overline{x})$ is computable Σ_α *if it is a c.e. disjunction of formulas $(\exists\overline{u})\,\psi(\overline{u},\overline{x})$, where ψ is a countable conjunction of formulas each of which is computable Σ_β or computable Π_β for some $\beta < \alpha$.*

  (ii) $\varphi(\overline{x})$ is computable Π_α *if it is a c.e. conjunction of formulas $(\forall\overline{u})\,\psi(\overline{u},\overline{x})$, where ψ is a countable disjunction of formulas each of which is computable Σ_β or computable Π_β for some $\beta < \alpha$.*

Again our formulas are in "normal" form, with negations occurring only next to atomic formulas. For each formula φ, the dual formula $neg(\varphi)$ is logically equivalent to the negation. Again it is not difficult to see that if φ is computable Σ_α, then $neg(\varphi)$ is computable Π_α, and vice versa.

In the result below, the definitions are as in the second approach.

Proposition 3.2. *Let $\mathcal{A}$ be an L-structure, and let $\varphi(\overline{x})$ be a computable infinitary L-formula. If $\varphi(\overline{x})$ is computable Σ_α (or computable Π_α), then the relation defined by $\varphi(\overline{x})$ in $\mathcal{A}$ is Σ^0_α (or Π^0_α) relative to $\mathcal{A}$.*

Proof. The computable Σ_0 and Π_0 formulas are the same as in the first definition. The computable Σ_1 and Π_1 formulas also turn out to be the same as in the first definition. Let $\varphi(\overline{x})$ be computable Σ_1 by the second definition. Then $\varphi(\overline{x})$ is the disjunction of a c.e. set of formulas $(\exists\overline{u}_i)\,\psi_i(\overline{x},\overline{u}_i)$, where each ψ_i is a *countable* conjunction of quantifier-free formulas of $L_{\omega_1\omega}$, so ψ_i is a quantifier-free formula of $L_{\omega_1\omega}$. Then $\varphi(\overline{x})$ is a computable Σ_1 formula according to the first definition.

We proceed by induction. Take $\alpha > 1$ and suppose that for $1 \leq \beta < \alpha$, the relation defined in $\mathcal{A}$ by a computable Σ_β formula is Σ^0_β relative to $\mathcal{A}$, and the relation defined by a computable Π_β formula is Π^0_β relative to $\mathcal{A}$, all uniformly. Let $\varphi(\overline{x})$ be a computable Σ_α formula, a c.e. disjunction of formulas $(\exists\overline{u}_i)\,\psi_i(\overline{x},\overline{u}_i)$, where ψ_i is a countable conjunction of formulas $\psi_{i,j}(\overline{x},\overline{u}_i)$, where $\psi_{i,j}$ is computable Π_β for some $\beta < \alpha$. For a tuple $\overline{a}$ in $\mathcal{A}$, $\mathcal{A} \models \varphi(\overline{a})$ iff there exist i and a tuple $\overline{b}_i$ such that for all of the countably many $\psi_{i,j}$, $\mathcal{A} \models \psi_{i,j}(\overline{a},\overline{b}_i)$. Using the oracle $\Delta^0_\alpha(\mathcal{A})$, we can apply a uniform procedure to decide whether $\mathcal{A} \models \psi_{i,j}(\overline{a},\overline{b}_i)$ for a single j, or for countably many. The relation defined by $\varphi(\overline{x})$ is c.e. relative to $\Delta^0_\alpha(\mathcal{A})$, so it is Σ^0_α relative to $\mathcal{A}$, according to the second definition. If $\varphi(\overline{x})$ is computable Π_α, we consider $neg(\varphi)$, the dual formula that is logically equivalent to the negation of φ. The relation defined by $neg(\varphi)$ is Σ^0_α relative to $\mathcal{A}$. The complementary relation is Π^0_α relative to $\mathcal{A}$, and this is the relation defined by our φ.

□

Let P be a computable propositional language, as above.

Definition 15 (Computable infinitary propositional formulas)**.**

- *φ is* computable Σ_0 *and* computable Π_0 *if it is a formula of P_{ω_1}.*

  For $\alpha > 0$,

  (i) φ is computable Σ_α *if it is a c.e. disjunction of countable conjunctions of formulas each of which is computable Σ_β or computable Π_β for some $\beta < \alpha$.*

  (ii) φ is computable Π_α *if it is a c.e. conjunction of countable disjunctions of formulas each of which is computable Σ_β or computable Π_β for some $\beta < \alpha$.*

Again, our formulas are in normal form. For each formula φ, the dual formula $neg(\varphi)$ is logically equivalent to the negation. If φ is computable Σ_α, then $neg(\varphi)$ is computable Π_α, and vice versa.

4 Relatively intrinsically arithmetical relations

Recall that all of our languages are computable, and when we are interested in complexity of a structure $\mathcal{A}$, we identify $\mathcal{A}$ with its atomic diagram. Thus, $\mathcal{A}$ is computable if its atomic diagram is computable.

We define "relatively intrinsically c.e." and "relatively intrinsically Σ^0_α" as in the standard setting, except that the terms "computable" and "c.e. relative to" are understood in the new sense.

Definition 16. *Let $\mathcal{A}$ be a computable structure, and let R be a relation on $\mathcal{A}$. We say that R is* relatively intrinsically Σ^0_α *on $\mathcal{A}$ if for all isomorphisms F from $\mathcal{A}$ onto a copy $\mathcal{B}$, $F(R)$ is Σ^0_α relative to $\mathcal{B}$.*

Below is the statement of our main result. There are really two different theorems, corresponding to the two different sets of definitions.

Theorem 4.1. *Let $1 \leq \alpha < \omega_1$. For a relation R on a computable structure $\mathcal{A}$ with domain assumed to be ω_1, the following are equivalent:*

1. *R is relatively intrinsically Σ^0_α on $\mathcal{A}$,*
2. *R is defined by a computable Σ_α formula.*

We give two proofs, one for each set of definitions. We begin with the first definition of the arithmetical hierarchy and the first definition of the computable infinitary formulas.

First proof. We get $2 \Rightarrow 1$ by Proposition 3.1. To prove that $1 \Rightarrow 2$, we use forcing, as in the standard setting. We suppose that $\mathcal{A}$ is a computable structure for a relational language L. We build a generic copy $\mathcal{B}$ with $F(\mathcal{B}) = \mathcal{A}$. Actually, we build a generic permutation F of ω_1, and we let $(\mathcal{B}, R') \cong_F (\mathcal{A}, R)$. The forcing conditions are countable partial permutations of ω_1. Note that the union of a countable chain of forcing conditions is a forcing condition.

We write $S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$, or $S_{(\Pi,\alpha,\gamma)}(\mathcal{B})$, for the set with index (Σ, α, γ), or (Π, α, γ), relative to $\mathcal{B}$. For $\alpha = 1$, $S_{(\Sigma,1,\gamma)}(\mathcal{B}) = W^{\mathcal{B}}_\gamma$, and $S_{(\Pi,1,\gamma)}(\mathcal{B})$ is the complementary set.

For our forcing language, we need formulas with the meanings below.

- $b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$
- $b \in S_{(\Pi,\alpha,\gamma)}(\mathcal{B})$
- $R' = S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$.

We use a propositional language. The propositional variables are the atomic sentences involving symbols from L, R, and constants from ω_1.

- For $\alpha = 1$:
  - To say that $b \in S_{(\Sigma,1,\gamma)}(\mathcal{B})$, we take the c.e. disjunction of the countable conjunctions of atomic formulas and negations of atomic formulas, which, if true of $\mathcal{B}$, would put $b \in W_\gamma^{\mathcal{B}}$—this is computable Σ_1. To be more formal, if U is the master enumeration program discussed in Proposition 1.2 and $D(\mathcal{B})$ is the atomic diagram for $\mathcal{B}$, we put $b \in W_\gamma^{\mathcal{B}}$ iff $\mathcal{B} \models \varphi = (\exists \sigma \subset \chi_{D(\mathcal{B})})[(\sigma, \gamma, b) \in U]$, where σ is countable and if $\sigma(\psi) = 1$, ψ is either atomic or the negation of an atomic sentence. Notice that φ can be written as a c.e. disjunction $\varphi = \bigvee_{\{\sigma \mid (\sigma,\gamma,b)\in U\}}(\bigwedge_{\sigma(\psi)=1} \psi \wedge \bigwedge_{\sigma(\psi)=0} \neg\psi)$, where the ψ are either atomic or the negations of atomic sentences and the conjunctions are countable. Since φ is the c.e. disjunction of a conjunction of computable Σ_0 formulas, it is computable Σ_1.
  - To say that $b \in S_{(\Pi,1,\gamma)}$, we apply *neg* to the formula saying that $b \in S_{(\Sigma,1,\gamma)}(\mathcal{B})$—this is computable Π_1.
- For $\alpha > 1$:
  - To say that $b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$, we take the c.e. disjunction over (Π, β, δ) in W_γ, where $\beta < \alpha$, of the formulas saying $b \in S_{(\Pi,\beta,\delta)}(\mathcal{B})$—this is computable Σ_α.

    We must be careful with limit ordinals, however, as our questions are bounded under the first set of definitions. We take the following c.e. disjunction over all $\beta < \alpha$:

    $$\bigvee_{1\le\beta<\alpha}\left[\bigvee_{\{(\Pi,\zeta,\delta)\in W_\gamma \ \& \ \zeta<\beta\}} \psi_{\zeta,\delta}\right]$$

    where the $\psi_{\zeta,\delta}$ is the formula stating $b \in S_{(\Pi,\zeta,\delta)}(\mathcal{B})$.
  - To say that $b \in S_{(\Pi,\alpha,\gamma)}(\mathcal{B})$, we apply *neg* to the formula saying that $b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$—this is computable Π_α.
- To say that $R' = S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$, we take the formula saying $\bigwedge_b (b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B}) \leftrightarrow Rb)$. This formula is computable $\Pi_{\alpha+1}$.

We let T include the computable Σ_β and Π_β formulas, for countable ordinals $\beta \le \alpha$, plus the $\Pi_{\alpha+1}$ formulas χ_γ saying that R' is equal to the set with index (Σ, α, γ) relative to $\mathcal{B}$, and the $\Sigma_{\alpha+1}$ formulas $neg(\chi_\gamma)$.

Definition 17 (Definition of forcing)**.** *Let p be a forcing condition.*

- *Suppose φ is computable Σ_0 and Π_0. Then $p \Vdash \varphi$ if the constants in the propositional variables that occur in φ are all in $dom(p)$ and p interprets these constants so as to make φ true in $(\mathcal{A}, R)$.*

- *Suppose φ is computable Σ_α, for $\alpha \geq 1$, say φ is the c.e. disjunction of formulas ψ_i, where each ψ_i is computable Π_β for some $\beta < \alpha$. Then $p \Vdash \varphi$ if $p \Vdash \psi_i$, for some i.*

- *Suppose φ is computable Π_α, for $\alpha \geq 1$, say φ is the c.e. conjunction of formulas ψ_i, where each ψ_i is computable Σ_β for some $\beta < \alpha$. Then $p \Vdash \varphi$ if for all i and all $q \supseteq p$, there is some $r \supseteq q$ such that $r \Vdash \psi_i$.*

We have the usual lemmas, extension, consistency, and density, all proved by induction on formulas in the forcing language.

Lemma 4.2 (Extension)**.** *If $p \Vdash \varphi$ and $q \supseteq p$, then $q \Vdash \varphi$.*

Lemma 4.3 (Consistency)**.** *It is not the case that $p \Vdash \varphi$ and $p \Vdash neg(\varphi)$.*

Lemma 4.4 (Density)**.** *For all p and φ, there exists $q \supseteq p$ such that q "decides" φ; i.e., $q \Vdash \varphi$ or $q \Vdash neg(\varphi)$.*

Definition 18 (Complete forcing sequence)**.** *A* complete forcing sequence*, or* c.f.s.*, is a sequence $(p_\alpha)_{\alpha<\omega_1}$ such that*

1. *if $\alpha < \beta$, then $p_\beta \supseteq p_\alpha$,*
2. *for all $\varphi \in T$, there is some α such that p_α decides φ,*
3. *for all $a \in \omega_1$, there is some α such that $a \in ran(p_\alpha)$.*

We can form a complete forcing sequence. For limit α, we let $p_\alpha = \cup_{\beta<\alpha} p_\beta$. Let $F = \cup_\alpha p_\alpha$ for $\alpha < \omega_1$. From this, we obtain $\mathcal{B}$ and R' such that $(\mathcal{B}, R') \cong_F (\mathcal{A}, R)$, as planned. Now, $\mathcal{B}$ and $(\mathcal{B}, R')$ are predicate structures. Taking the positive sentences in the atomic diagrams, we obtain corresponding propositional structures, which we call by the same names.

Lemma 4.5 (Truth and Forcing Lemma)**.** *For $\varphi \in T$, $(\mathcal{B}, R') \models \varphi$ iff there is some α such that $p_\alpha \Vdash \varphi$.*

Proof. We proceed by induction on φ.

(1) Suppose φ is computable Σ_0 and computable Π_0. Then $(\mathcal{B}, R') \models \varphi$ if and only if for all α such that $dom(p_\alpha)$ includes the constants appearing in φ, $p_\alpha \Vdash \varphi$. So, the statement holds.

(2) Suppose φ is computable Σ_β for $\beta \geq 1$. Then φ is a c.e. disjunction $\bigvee_i \varphi_i$, where the statement holds for each φ_i. We have

$$\begin{aligned}(\mathcal{B}, R') \models \varphi &\Leftrightarrow (\exists i)[(\mathcal{B}, R') \models \varphi_i] \\ &\Leftrightarrow (\exists i)(\exists n)[p_n \Vdash \varphi_i] \\ &\Leftrightarrow (\exists n)[p_n \Vdash \varphi].\end{aligned}$$

(3) Finally, suppose that φ is computable Π_β for $\beta \geq 1$. Then φ is the c.e. conjunction $\bigwedge_i \varphi_i$, where the statement holds for each φ_i. First, suppose that $(\mathcal{B}, R') \models \varphi$. Then $(\mathcal{B}, R') \models \varphi_i$, for all i. By Lemma 4.4, we know that there is an n such that p_n decides φ. Suppose $p_n \Vdash neg(\varphi)$. Then for some i, $p_n \Vdash neg(\varphi_i)$. By the induction hypothesis, there is an m such that $p_m \Vdash \varphi_i$. By Lemma 4.2, we may assume that $m = n$ if we just take the larger. Then Lemma 4.3 gives us a contradiction. So $p_n \Vdash \varphi$. Now suppose that $p_n \Vdash \varphi$. Then for each i, there is some $m \geq n$ such that p_m decides φ_i. $p_n \Vdash \varphi$ implies that there is a $q \supseteq p_m$ such that $q \Vdash \varphi_i$. So $p_m \Vdash \varphi_i$. By the induction hypothesis, $(\mathcal{B}, R') \models \varphi_i$ for all i. Therefore, $(\mathcal{B}, R') \models \varphi$.

□

We have definability of forcing. We are especially interested in the computable Σ_α and computable Π_α formulas that do not involve R. Let T' be the set of formulas in T that do not involve R. Forcing for these formulas is definable in $\mathcal{A}$.

Lemma 4.6 (Definability of forcing)**.** *For any $\varphi \in T'$, and for any $\bar{b}$ and $\bar{x}$ of the same countable ordinal arity, there is a predicate formula $Force_{\bar{b},\varphi}(\bar{x})$ such that $\mathcal{A} \models Force_{\bar{b},\varphi}(\bar{a})$ iff the correspondence taking b_i to a_i is a forcing condition p such that $p \Vdash \varphi$. Moreover, if φ is computable Σ_β, or Π_β, for $\beta \leq \alpha$, then $Force_{\bar{b},\varphi}(\bar{x})$ is also computable Σ_β, or Π_β.*

Proof. We proceed by induction on φ.

(1) Suppose φ is computable Σ_0 and computable Π_0. Given a countable tuple $\bar{b}$ and a formula φ from P_{ω_1}, we can effectively determine whether $\bar{b}$ includes all of the elements that appear in the propositional variables of φ. We take $Force_{\bar{b},\varphi}(\bar{x})$ to be $\bot$ if this is not true, and otherwise, it is essentially the result of replacing b_i by x_i in the formula φ, and thinking of the result as a predicate formula. Thus, the statement holds.

(2) Suppose φ is Σ_β. Then φ is a c.e. disjunction of formulas $\bigvee_i \varphi_i$, where each φ_i is computable Π_{ζ_i} for some $\zeta_i < \beta$, and we have determined $Force_{\bar{b},\varphi_i}(\bar{x})$ for all i. Define the c.e. disjunction

$$Force_{\bar{b},\varphi}(\bar{x}) = \bigvee_i Force_{\bar{b},\varphi_i}(\bar{x}).$$

Thus, the statement holds. Furthermore, as each $Force_{\bar{b},\varphi_i}(\bar{x})$ is computable Π_{ζ_i} for $\zeta_i < \beta$, by the induction hypothesis, we have that $Force_{\bar{b},\varphi}(\bar{x})$ is computable Σ_β.

(3) Suppose φ is computable Π_β. Then φ is the c.e. conjunction of formulas $\bigwedge_i \varphi_i$, where each φ_i is computable Σ_{ζ_i} for some $\zeta_i < \beta$, and we have determined $Force_{\bar{b},\bar{d},\varphi_i}(\bar{x}, \bar{y})$ for all i and all $\bar{d}$. From the definition of

forcing, we have that $p \Vdash \varphi \Leftrightarrow \forall i \forall q \supseteq p\ \exists r \supseteq q\ r \Vdash \varphi_i$. We define the *predicate* c.e. conjunction

$$Force_{\overline{b},\varphi}(\overline{x}) = \bigwedge_i \bigwedge_{\overline{d}} \forall \overline{y} \bigvee_{\overline{d_1}} \exists \overline{z}\ Force_{\overline{b},\overline{d},\overline{d_1},\varphi_i}(\overline{x},\overline{y},\overline{z}),$$

where all the tuples among the $\overline{d},\overline{y},\overline{d_1}$ are all distinct, and the length of the variables $\overline{y},\overline{z}$ match the corresponding tuples of constants $\overline{d},\overline{d_1}$, which come from the domains of q and r respectively. Then the statement holds, and by the induction hypothesis, we note that the formula is computable Π_β.

□

We are ready to complete the proof of the theorem. By assumption and Lemma 4.5, p forces $S_{(\Sigma,\alpha,\gamma)}(\mathcal{B}) = R'$, for some γ. Say p maps $\overline{d}$ to $\overline{c}$. We can see that $a \in R$ iff there is some $q \supseteq p$ such that $q(b) = a$ and $q \Vdash b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$. We have a computable Σ_α predicate formula $\varphi(\overline{c}, x)$ saying that there exists $q \supseteq p$ such that $q(b) = x$ and $q \Vdash b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$. We take the disjunction over $b, \overline{b}_1$ of the formulas $(\exists \overline{u})\, Force_{\overline{d},b,\overline{b}_1,b\in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})}(\overline{c}, x, \overline{u})$. This formula defines R, and it is straightforward to see that it is computable Σ_α.

□

We have completed the first proof, using the first set of definitions. We now prove the same statement using the second set of definitions.

Second proof. Again we get $2 \Rightarrow 1$ by Proposition 3.2. To prove that $1 \Rightarrow 2$, we use forcing. Most of the proof is the same. When we write $S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$, we mean the set $W_\gamma^{\Delta^0_\alpha(\mathcal{B})}$. We write $S_{(\Pi,\alpha,\gamma)}(\mathcal{B})$ for the complementary set. Recall that $\Delta^0_1(\mathcal{B})$ is just $\mathcal{B}$, which we have identified with its atomic diagram. $\Delta^0_{\alpha+1}(\mathcal{B})$ is the jump of $\Delta^0_\alpha(\mathcal{B})$, and for limit α, $\Delta^0_\alpha(\mathcal{B})$ is the set of pairs (β, b) such that $\beta < \alpha$ and $b \in \Delta^0_\alpha(\mathcal{B})$.

For our forcing language, we need formulas with the meanings below.

1. $b \in \Delta^0_\beta(\mathcal{B})$, for $\beta \leq \alpha$
2. $b \notin \Delta^0_\beta(\mathcal{B})$, for $\beta \leq \alpha$
3. $b \in S_{(\Sigma,\beta,\gamma)}(\mathcal{B})$, for $\beta \leq \alpha$
4. $b \in S_{(\Pi,\beta,\gamma)}(\mathcal{B})$, for $\beta \leq \alpha$
5. $R' = S_{(\Sigma,\beta,\gamma)}(\mathcal{B})$.

Again, the propositional variables are the atomic sentences involving symbols from L, R, and constants from ω_1. We identify propositional variables, or negations of propositional variables, with their codes. The set of ordinals that are codes for propositional variables or negations is computable. For the first four items above, we do not use the extra symbol R.

We proceed by induction on β, starting with $\beta = 1$.

- If b is an atomic sentence in the language of $\mathcal{B}$, we write b to say that b is in the atomic diagram of $\mathcal{B}$. Similarly, if b is the negation of an atomic sentence, we write $\neg b$ to say that b is in the atomic diagram.
- If b does not have either of these forms, then we write $\bot$, reflecting the fact that b cannot be in the atomic diagram.
- To say that b is not in the atomic diagram of $\mathcal{B}$, we write $\neg b$ in case b is an atomic sentence, b in case b is the negation of an atomic sentence, and $\top$ if b does not have either of these forms.
- To say that $b \in S_{(\Sigma,1,\gamma)}(\mathcal{B})$, we take the c.e. disjunction of countable conjunctions of propositional variables and negations, representing information which, if true of $\mathcal{B}$, would put b into $W_\gamma^{\mathcal{B}}$. This is computable Σ_1.
- To say that $b \in S_{(\Pi,1,\gamma)}(\mathcal{B})$, we apply *neg* to item 3. So, it is computable Π_1.

Suppose we have the desired formulas for β, and consider $\beta + 1$. In this case, $\Delta^0_{\beta+1}(\mathcal{B})$ is the jump of $\Delta^0_\beta(\mathcal{B})$, which is Σ^0_β, so we already have a formula saying $b \in \Delta^0_{\beta+1}(\mathcal{B})$. We also have a formula saying $b \notin \Delta^0_{\beta+1}(\mathcal{B})$.

- To say that $b \in S_{(\Sigma,\beta+1,\gamma)}(\mathcal{B})$, we take the c.e. disjunction of the countable conjunctions of statements which, if true of $\Delta^0_{\beta+1}(\mathcal{B})$, would put $b \in W_\gamma^{\Delta^0_{\beta+1}(\mathcal{B})}$.
- To say that $b \in S_{(\Pi,\beta+1,\gamma)}(\mathcal{B})$, we take the dual formula, equivalent to the negation.

Suppose β is a limit ordinal, and we have the formulas for smaller ordinals. In this case, $\Delta^0_\beta(\mathcal{B})$ is the direct sum of $\Delta^0_{\beta'}(\mathcal{B})$, for $\beta' < \beta$.

- For each b that is the code for a pair (β', x), we write $x \in \Delta^0_{\beta'}(\mathcal{B})$ to say that $b \in \Delta^0_\beta(\mathcal{B})$, and we write $x \notin \Delta^0_{\beta'}(\mathcal{B})$ to say that $b \notin \Delta^0_\beta(\mathcal{B})$. If b is not of this form, we write $\top$ to say that $b \notin \Delta^0_\beta(\mathcal{B})$, and we write $\bot$ to say that $b \in \Delta^0_\beta(\mathcal{B})$.
- To say that $b \in S_{\Sigma,\beta,\gamma)}(\mathcal{B})$, we take the c.e. disjunction of countable conjunctions of statements which, if true of $\Delta^0_\beta(\mathcal{B})$, would but $binW_\gamma^{\Delta^0_\beta(\mathcal{B})}$.
- To say that $b \in S_{\Pi,\beta,\gamma)}(\mathcal{B})$, we take the dual formula.

Finally, to say that $R' = S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$, we take the formula saying $\bigwedge_b (b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B}) \leftrightarrow R'b)$. This is computable $\Pi_{\alpha+1}$.

We let T include the computable Σ_β and Π_β formulas, for countable ordinals $\beta \leq \alpha$, plus the $\Pi_{\alpha+1}$ formulas χ_β saying that $R' = S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$ and the $\Sigma_{\alpha+1}$ formulas $neg(\chi_\beta)$. We define forcing for the formulas in T.

Definition 19 (Definition of forcing). *Let p be a forcing condition.*

- *Suppose φ is computable Σ_0 and Π_0. Then $p \Vdash \varphi$ if the constants in the propositional variables that occur in φ are all in $dom(p)$ and p interprets these constants so as to make φ true in $(\mathcal{A}, R)$.*
- *Suppose φ is computable Σ_α, for $\alpha \geq 1$, say φ is the c.e. disjunction of formulas ψ_i, where each ψ_i is a countable conjunction of formulas each of which is Σ_β or Π_β for some $\beta < \alpha$. Then $p \Vdash \varphi$ if $p \Vdash \psi_i$, for some i.*
- *Suppose φ is computable Π_α, for $\alpha \geq 1$, say φ is the c.e. conjunction of formulas ψ_i, where each ψ_i is the countable disjunction of formulas each of which is Π_β or Σ_β for some $\beta < \alpha$. Then $p \Vdash \varphi$ if for all i and all $q \supseteq p$, there is some $r \supseteq q$ such that $r \Vdash \psi_i$.*

We have the usual lemmas, extension, consistency, and density, all proved by induction on formulas in the forcing language.

Lemma 4.7 (Extension). *If $p \Vdash \varphi$ and $q \supseteq p$, then $q \Vdash \varphi$.*

Lemma 4.8 (Consistency). *It is not the case that $p \Vdash \varphi$ and $p \Vdash neg(\varphi)$.*

Lemma 4.9 (Density). *For all p and φ, there exists $q \supseteq p$ such that q "decides" φ; i.e., $q \Vdash \varphi$ or $q \Vdash neg(\varphi)$.*

We can form a complete forcing sequence F. For limit α, we let $p_\alpha = \cup_{\beta<\alpha} p_\beta$. Let $F = \cup_\alpha p_\alpha$ for $\alpha < \omega_1$. From this, we obtain $\mathcal{B}$ and R' such that $(\mathcal{B}, R') \cong_F (\mathcal{A}, R)$, as planned. Now, $\mathcal{B}$ and $(\mathcal{B}, R')$ are predicate structures. Taking the positive sentences in the atomic diagrams, we obtain corresponding propositional structures, which we call by the same names.

Lemma 4.10 (Truth and Forcing Lemma). *For $\varphi \in T$, $(\mathcal{B}, R') \models \varphi$ iff there is some α such that $p_\alpha \Vdash \varphi$.*

Proof. The proof looks exactly the same as for the first set of definitions, but of course, the formulas are different.

□

We have definability of forcing. We are especially interested in the computable Σ_α and computable Π_α formulas that do not involve R. Let T' be the set of formulas in T that do not involve R. Forcing for these formulas is definable in $\mathcal{A}$.

Lemma 4.11 (Definability of forcing). *For any $\varphi \in T'$, and for any $\overline{b}$ and $\overline{x}$ of the same countable ordinal arity, there is a predicate formula $Force_{\overline{b},\varphi}(\overline{x})$ such that $\mathcal{A} \models Force_{\overline{b},\varphi}(\overline{a})$ iff the correspondence taking b_i to a_i is a forcing condition p such that $p \Vdash \varphi$. Moreover, if φ is computable Σ_β, or Π_β, for $1 \leq \beta \leq \alpha$, then $Force_{\overline{b},\varphi}(\overline{x})$ is also computable Σ_β, or Π_β.*

Proof. The proof is the same as with the first set of definitions, but of course, the formulas are different. □

We are ready to complete the proof of the theorem. Suppose p forces $S_{(\Sigma,\alpha,\gamma)}(\mathcal{B}) = R'$, where p maps $\overline{d}$ to $\overline{c}$. We can see that $a \in R$ iff there is some $q \supseteq p$ such that $q(b) = a$ and $q \Vdash b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$. We have a computable Σ_α predicate formula $\varphi(\overline{c}, x)$ saying that there exists $q \supseteq p$ such that $q(b) = x$ and $q \Vdash b \in W_\alpha^{\mathcal{B}}$. We take the disjunction over $b, \overline{b}_1$ of the formulas $(\exists \overline{u})\, Force_{\overline{d},b,\overline{b}_1,b\in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})}(\overline{c}, x, \overline{u})$. This formula defines R.

□

We have completed the second proof.

5 Which definition is best?

The two definitions of the arithmetical hierarchy are not identical. We have seen that they are not equivalent. We would like to choose one of the definitions as in some way better. We cannot say that one definition is more productive. For both definitions, we have a result saying that a relation is relatively intrinsically Σ^0_α iff it is definable by a computable Σ_α formula. It seems, at least to the second and fourth authors, that the second definition is more natural. In the standard setting, an element enters a Σ^0_5 set based on finitely much Δ^0_5 information. Similarly, in our second definition, an element enters a Σ^0_ω set based on countably much information about Δ^0_ω. The information may use the full power of the Δ^0_ω oracle—there may be questions about Δ^0_n for all n. Under the first definition, an element enters a Σ^0_ω set based on countably much Δ^0_n information, for some n.

A further confirmation that the second definition is more natural is the connection to much earlier work of Jensen [6]. As part of his analysis of the constructible hierarcy, Jensen defined classes of sets Σ^0_n over L_β for all ordinals β. We are grateful to Sy Friedman for explaining to us the connection between our classes and Jensen's. Our Σ^0_1 sets are, in Jensen's hierarchy, Σ_1 over L_{ω_1}, our Σ^0_2 sets are Σ_2 over L_{ω_1}, and in general, our Σ^0_n sets are Σ_n over L_{ω_1}. Our Σ^0_ω sets are, in Jensen's hierarchy, Σ_1 over L_{ω_1+1}, our $\Sigma^0_{\omega+n}$ sets are Σ_{1+n} over L_{ω_1+1}, and so on. The oracle we call Δ^0_2, a complete Σ^0_1 set, is Jensen's master code in n for Our Δ^0_α oracles are Jensen's Δ^0_n "master codes" at various levels of L. Friedman [5] considered a version of Jensen's hierarchy with ω_1 replaced by $\aleph_{omega_1}$.

References

[1] N. Greenberg and J. F. Knight, "Computable structure theory in the setting of ω_1, paper for Proceedings of first EMU workshop.

[2] C. J. Ash, J. F. Knight, M. Mannasse, and T. Slaman, "Generic copies of countable structures", *Annals of Pure and Appl. Logic*, vol. 42(1989), pp. 195-205.

[3] C. J. Ash, J. F. Knight, *Computable Structures and the Hyperarithmetical Hierarchy*, Studies of Logic and the Foundations of Mathematics, vol. 144 (2000).

[4] J. Chisholm, "Effective model theory versus recursive model theory", *J. Symb. Logic*, vol. 55(1990), pp. 1168-1191.

[5] S.-D. Friedman, "Negative Solutions to Post's Problem II", *Annals of Mathematics*, Vol. 113(1981), pp. 25-43.

[6] R. Jensen, "The fine structure of the constructible hierarchy", *Annals of Mathematical Logic*, 1972.

[7] M. Vanden Boom, "The effective Borel hierarchy", *Fund. Math.*, vol. 195(2007), pp. 269-289.