2013

The arithmetical hierarchy in the setting of ω_1

Jacob Carson

Jesse Johnson

Julia F. Knight

Karen Lange
klange2@wellesley.edu

Charles McCoy CSC

See next page for additional authors

Recommended Citation

The arithmetical hierarchy in the setting of ω_1 computability, J. Carson, J. Johnson, J. Knight, K. Lange, C. McCoy, and J. Wallbaum, Computability, 2(2) (2013), 93-105.

Authors
Jacob Carson, Jesse Johnson, Julia F. Knight, Karen Lange, Charles McCoy CSC, and John Walbaum

The arithmetical hierarchy in the setting of ω_1

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

April 29, 2013

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 (not just the finite levels). We consider two different ways of doing this—one based on the standard definition of the hyperarithmetical hierarchy, and the other based on the standard definition of the effective Borel hierarchy. For each definition, we define computable infinitary formulas through all countable levels, and we obtain analogues of the well-known results from [2] and [4] saying that a relation is relatively intrinsically Σ^0_α just in case it is definable by a computable Σ_α formula. Although we obtain the same results for the two definitions of the arithmetical hierarchy, we conclude that the definition resembling the standard definition of the hyperarithmetical hierarchy seems preferable.

1 Introduction

We consider computability in the setting of ω_1. The countable ordinals play the role of natural numbers, and countable sets play the role of finite sets. We assume $V = L$. This implies that all reals are present in L_{ω_1}. In fact, every subset of ω_1 is "amenable" for L_{ω_1}; i.e., for all $A \subseteq L_{\omega_1}$ and all $x \in L_{\omega_1}$, $A \cap x \in L_{\omega_1}$. In the remainder of the introduction, we review some basic definitions and results from [1]. In Section 2, we define the arithmetical hierarchy—the Σ^0_α and Π^0_α subsets of ω_1, for countable ordinals α. We do this in two different ways. The first is based on the standard definition of the hyperarithmetical hierarchy—a set is Σ^0_α if it is c.e. relative to a complete Δ^0_α oracle. The second definition is based on the definition of the effective Borel hierarchy—a set is Σ^0_α if it is a c.e. union of sets each of which is Π^0_β for some $\beta < \alpha$.

In Section 3, we define computable infinitary Σ_α formulas in two different ways, corresponding to the different definitions of the arithmetical hierarchy. We allow countable tuples of quantifiers, and our formulas will be from L_{ω_2,ω_1}. In Section 4, we give the main results, saying that for each set of definitions, 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. This is the analogue of a result for the standard computability setting in [2] and [4]. Finally, in Section 5, we explain why the first set of definitions seems preferable.

1.1 Basic definitions

Below, we first say what it means for a set or relation on L_{ω_1} 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* bounded*, or* Δ_0*, if it is defined by a finitary formula with only bounded quantifiers,* $(\exists x \in y)$ *and* $(\forall x \in y)$*, in the language with* $\in$ *and constants from* L_{ω_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* $\varphi(\overline{c}, x)$*, with finitely many parameters—the formula is finitary, with only existential and bounded quantifiers, and all negations appear inside quantifiers.*
- *A relation* $R \subseteq (L_{\omega_1})^n$ *is* computable *if it and its complement 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, and that computable functions are generally defined by recursion on ordinals—the Σ_1 definition for a function f says that there exists 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.

Results of Gödel give 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 prvides 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 more details on this point, see [1].

Above, we were thinking of relations and functions of finite arity. We may allow relations and functions of arity α, where $\alpha < \omega_1$. 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—the set of ordinal codes for sequences in* R *is c.e.*
- *A relation of arity* α *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 $\{\overline{a}, f(\overline{a}) : \overline{a} \in domf\}$ is a c.e. relation.*

- We have a c.e. set C of codes for pairs $(\varphi, \overline{c})$, 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. *For $\alpha < \omega_1$, 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.

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. After all, in the current setting, countable sets, such as L_β, appear "finite". In the standard setting, the class of c.e. sets is closed under finite intersection. We have the analogue here.

Proposition 1.1.
The class of c.e. sets is closed under countable intersection.

Proof. Let Γ be a countable set of countable ordinals (indices for c.e. sets), and let $S = \cap_{\gamma \in \Gamma}$. We have $x \in S$ iff $(\exists \beta)\,(\forall \gamma \in \Gamma)\, x \in W_{\gamma,\beta}$, so S is Σ_1 definable in L_{ω_1}.

□

1.2 Relative computability and jumps

Proposition 1.2. *There is a c.e. set $U \subseteq \omega_1 \times \omega_1$ consisting of the pairs (α, β) such that $\beta \in W_\alpha$.*

In the standard setting, we define the halting set K, and we prove that it is c.e. and non-computable. We then use the $s-m-n$ theorem to show that all c.e. sets are 1-reducible to K. In the setting of ω_1, we could proceed in exactly the same way [1], letting $K = \{\alpha : \alpha \in W_\alpha\}$. Instead, we define a set that is obviously 1-complete.

Definition 5. *Let $K = \{(\alpha, y) : y \in W_\alpha\}$.*

This set K is c.e. We have a Σ_1 definition for K, saying that there exists β such that L_β contains the pair $h(\alpha) = (\varphi(\overline{u}, y), \overline{a})$, and $L_\beta \models \varphi(\overline{a}, y)$. The complement of K cannot be c.e., for then the set $\{\alpha : \alpha \notin W_\alpha\}$ would also be c.e.

Relative computability is important in what follows.

Definition 6. *Let $X \subseteq \omega_1$.*

- *A relation is* c.e. relative to X *if it is* Σ_1*-definable in* $(L_{\omega_1}, \in, X)$.
- *A relation is* computable relative to X *if it and its complement 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 1.3. *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 of the proposition above appears in [1].

We define the *jump* of a set $X \subseteq \omega_1$ to make the universality obvious.

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

As in the standard setting, for each X, X' is c.e. relative to X and not computable relative 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 $x \in X^{(\beta)}$.

It is convenient to have, for each countable ordinal $\alpha \geq 1$, a name for a specific oracle set.

Notation. For finite $n \geq 1$, we write Δ_n^0 for $\varnothing^{(n-1)}$, and for countable ordinals $\alpha \geq \omega$, we write Δ_α^0 for $\varnothing^{(\alpha)}$. We can relativize to a set X. For finite $n \geq 1$, we let $\Delta_n^0(X) = X^{(n-1)}$, and for $\alpha \geq \omega$, we let $\Delta_\alpha^0(X) = X^{(\alpha)}$.

2 The arithmetical hierarchy

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

2.1 First definition

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

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

- *R is Σ^0_0 and Π^0_0 if it is computable.*
- *For a countable ordinal $\alpha > 0$,*
  - *R is Σ^0_α if it is c.e. relative to Δ^0_α,*
  - *R is Π^0_α if the complementary relation $\neg R$ is Σ^0_α.*

We may relativize this.

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

- *R is $\Sigma^0_0(X)$ and $\Pi^0_0(X)$ if R is computable relative to X.*
- *For a countable ordinal $\alpha > 0$,*
  - *R is $\Sigma^0_\alpha(X)$ if it is c.e. relative to $\Delta^0_\alpha(X)$,*
  - *R is $\Pi^0_\alpha(X)$ if the complementry relation $\neg R$ is $\Sigma^0_\alpha(X)$*

.

We assign indices to the Σ^0_α and Π^0_α sets, for $\alpha \geq 1$. We ignore the case where $\alpha = 0$. 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. 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. When we relativize to a set X, we use the same indices. The set with index (Σ, α, γ) relative to X is $W_\gamma^{\Delta^0_\alpha(X)}$, and the set with index (Π, α, γ) relative to X is the complement.

2.2 Second definition

In our second definition for the arithmetical hierarchy, we follow the approach used in defining the effective Borel hierarchy [6].

Definition 11. *Let R be a relation on ω_1.*

- *R is Σ^0_0 and Π^0_0 if R is computable.*
- *R is Σ^0_1 if it is c.e., R is Π^0_1 if the complement, $\neg R$, is c.e.*
- *For countable ordinal $\alpha > 1$,*
  - *R is Σ^0_α if it is a c.e. union of relations, each of which is Π^0_β for some $\beta < \alpha$,*

– *R is Π^0_α if $\neg R$ is Σ^0_α.*

We may relativize to X in a straightforward way. A relation is Σ^0_0 and Π^0_0 relative to X if it is computable relative to X. For a countable ordinal $\alpha > 0$, a relation R is Σ^0_α relative to X if it is a c.e. union of relations, each of which is Π^0_β relative to X for some $\beta < \alpha$; R is Π^0_α relative to X if $\neg R$ is Σ^0_α relative to X.

We may assign indices to the Σ^0_α and Π^0_α sets in a natural way. We ignore $\alpha = 0$. For $\alpha = 1$, $(\Sigma, 1, \gamma)$ is the index for the c.e. set W_γ, and $(\Pi, 1, \gamma)$ is the index for the complementary set. For $\alpha > 1$, (Σ, α, γ) is the union of the sets with indices in W_γ of the form (Π, β, δ), where $1 \leq \beta < \alpha$. Similarly, (Π, α, γ) is the index for the intersection of the sets with indices in W_γ of the form (Σ, β, δ), for $1 \leq \beta < \alpha$.

When we relativize to a set X, we use the same indices. The set with index $(\Sigma, 1, \gamma)$ relative to X is W^X_γ; the set with index $(\Pi, 1, \gamma)$ relative to X is the complement. For $\alpha > 1$, the set with index (Σ, α, γ) relative to X is the union of the sets with indices (relative to X) in W_γ of the form (Π, β, δ), where $1 \leq \beta < \alpha$; the set with index (Π, α, γ) relative to X is the intersection of the sets with indices (relative to X) in W_γ of the form (Σ, β, δ), where $1 \leq \beta < \alpha$.

2.3 Comparing the two definitions

We write $\Sigma^0_\alpha(I)$, $\Pi^0_\alpha(I)$ for the first definition, and $\Sigma^0_\alpha(II)$, $\Pi^0_\alpha(II)$ for the second.

Proposition 2.1. *For finite n, a set or relation is $\Sigma^0_n(I)$ iff it is $\Sigma^0_n(II)$.*

Proof. Under both definitions, the Σ^0_0 and Π^0_0 sets and relations are the computable ones. Also, under both definitions, the Π^0_n sets are the complements of the Σ^0_n sets. A set is $\Sigma^0_1(I)$ iff it is c.e. relative to $\varnothing$, and a set is $\Sigma^0_1(II)$ iff it is c.e. These are clearly equivalent. For larger n, we use some approximations to $\varnothing'$ and $\varnothing^{(k)}$ for $k < n$.

The set $\varnothing'$ is c.e., so we have a formula γ_1, with only bounded quantifiers, such that $y \in \varnothing'$ iff the formula $(\exists \overline{v})\gamma_1(\overline{v}, y)$ holds in L_{ω_1}. (We ignore the parameters.) For an ordinal β, let $Y_{1,\beta}$ be the set of $y \in L_\beta$ such that $(\exists \overline{v})\gamma_1(\overline{v}, y)$ holds in L_β; i.e., we can take $\overline{v} \in L_\beta$. The relation $Y = Y_{1,\beta}$ (on Y and β) is computable. We say that β is 1-*good* if $Y_{1,\beta} = \varnothing' \cap L_\beta$. This means that for all $y \in L_\beta$, if $y \in \varnothing'$, then $(\exists \overline{v})\gamma_1(\overline{v}, y)$ holds in L_β. The set of β that are 1-good is Π^0_1.

Similarly, $\varnothing^{(n+1)}$ is c.e. relative to $\varnothing^n$, so we have a formula γ_{n+1}, with only bounded quantifiers, such that $y \in \varnothing^{(n+1)}$ iff the formula $(\exists \overline{v})\gamma(\overline{v}, x)$ holds in $(L_{\omega_1}, \varnothing^{(n)})$. (Again we ignore the parameters.) Let $Y_{n+1,\beta}$ be the set of $y \in L_\beta$ such that $(\exists \overline{v})\gamma_{n+1}(\overline{v}, y)$ holds in $(L_\beta, Y_{n,\beta})$. The relation $Y = Y_{n+1,\beta}$ is computable. We say that β is *(n+1)-good* if it is k-good for all $k \leq n$ and $Y_{n+1,\beta} = \varnothing^{(n+1)} \cap \beta$. This means that for all $y \in L_\beta$, if $y \in \varnothing^{(n+1)}$, then $(\exists \overline{v})\gamma_{n+1}(\overline{v}, y)$ holds in $(L_\beta, Y_{n,\beta})$. The set of β that are $(n+1)$-good is Π^0_{n+1}.

Assuming that the two definitions agree at level n, where $n \geq 1$, we show that they agree at level $n+1$. Suppose R is $\Sigma^0_{n+1}(I)$, so we have a formula δ, with only bounded quantifiers, such that $x \in R$ iff the formula $(\exists \overline{u})\delta(\overline{u}, x)$ holds in $(L_{\omega_1}, \varnothing^{(n)})$. We have $x \in R$ iff there is some β such that $x \in L_\beta$, β is n-good, and $(\exists \overline{u}, \delta(\overline{u}, x)$ holds in $(L_\beta, Y_{n,\beta})$. For each ordinal α, let R_α be the set of x such that for some $\beta < \alpha$, β is n-good, $x \in L_\beta$, and $(\exists \overline{u})\delta(\overline{u}, x)$ holds in $(L_\beta, Y_{n,\beta})$. The sets R_α are Π^0_n, uniformly in α, and R is a c.e. union. Therefore, R is $\Sigma^0_{n+1}(II)$.

Now, suppose R is $\Sigma^0_{n+1}(II)$. Say R is the union of Π^0_n sets R_i with indices i in a c.e. set I. Then $x \in R$ iff there exists i such that $i \in I$ and x is in the Π^0_n set with index i. The set of all pairs (x, i) such that x is an element of the Π^0_n set with index i is Π^0_n, so it is computable relative to $\varnothing^{(n)}$ (think of the complementary sets). Therefore, R is c.e. relative to $\varnothing^{(n)}$, so it is $\Sigma^0_{n+1}(I)$.

□

The two definitions disagree at level ω and beyond. Under the first definition, the computation that puts a particular element into a Σ^0_ω may involve Δ^0_n information for all n. Under the second definition, an element enters a Σ^0_ω by entering some Π^0_n set.

Proposition 2.2. *There is a set S that is $\Sigma^0_\omega(I)$ and $\Pi^0_\omega(I)$ but not $\Sigma^0_\omega(II)$.*

Proof. Each set that is $\Sigma^0_\omega(II)$ has an index of the form (Σ, ω, α)—the set is equal to the union of the sets that have indices in W_α of the form (Π, n, β), for $n \in \omega$. We can define a set S that is $\Sigma^0_\omega(I)$ and $\Pi^0_\omega(I)$, such that $\alpha \in S$ iff α is not in the set with second-definition index (Σ, ω, α); we define S to diagonalize out of the class of $\Sigma^0_\omega(II)$ sets. For each α and n, let $S_{(\alpha,n)}$ be the union of the Π^0_k sets with indices in W_α of the form (Π, k, β), with $k < n$ 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 $\Sigma^0_\omega(II)$ set with index (Σ, ω, α). For each countable α, we can determine, computably relative to Δ^0_ω, whether $(\forall n \in \omega)$, $(\alpha \notin S_{(\alpha,n)})$. (The quantifier $(\forall n \in \omega)$ is bounded.) We let $\alpha \in S$ iff $(\forall n \in \omega)\, \alpha \notin S_{(\alpha,n)}$. Then S is $\Delta^0_\omega(I)$, but it is not equal to any of the $\Sigma^0_\omega(II)$ sets.

□

The set S is $\Sigma^0_{\omega+1}(II)$. The two hierarchies differ by a jump at level ω. They remain off by a jump all the way up.

3 Computable infinitary formulas

In the standard setting of computability, formulas of $L_{\omega_1,\omega}$ are infinitary formulas in which the infinite disjunctions and conjunctions are over countable sets, but there is no infinite nesting of quantifiers. We consider predicate formulas with a finite tuple of free variables. There is no prenex normal form for these formulas—in general, we cannot bring quantifiers to the front. However, we can bring negations inside, and this results in a kind of normal form. We classify formulas of $L_{\omega_1,\omega}$ in normal form as Σ_α or Π_α for countable ordinals α. Computable infinitary formulas are formulas of $L_{\omega_1,\omega}$ in which the disjunctions and conjunctions are over c.e. sets. We classify the computable infinitary formulas as computable Σ_α or computable Π_α for computable ordinals α. For more information on computable infinitary formulas in the standard setting, see [3].

In this section, we give definitions, for the setting of ω_1, of the computable Σ_α and computable Π_α formulas for countable ordinals α. Of course, c.e. disjunctions and conjunctions may be uncountable. We allow countable tuples of variables. Thus, our computable infinitary formulas are formulas of L_{ω_2,ω_1}, not $L_{\omega_1,\omega}$.

We give two different definitions, corresponding to our two different definitions of the arithmetical hierarchy.

3.1 First definition

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

Definition 12 (Computable infinitary predicate formulas)**.** *Let L be a predicate language. For simplicity, we suppose that the symbols are the usual kind, with* finite *arity. We suppose that the set of symbols in L is computable, and that the function that assigns the type (relation, operation) and arity to symbols in L is computable. We consider L-formulas $\varphi(\overline{x})$ with a countable tuple of variables $\overline{x}$.*

- $\varphi(\overline{x})$ *is* computable Σ_0 *and* computable Π_0 *if it is a quantifier-free formula of $L_{\omega_1,\omega}$—we allow a countable tuple of variables.*
- *For $\alpha > 0$,*
  - $\varphi(\overline{x})$ *is* computable Σ_α *if it is a c.e. disjunction of formulas $(\exists\overline{u})\,\psi(\overline{u},\overline{x})$, where $\overline{u}$ is a countable tuple of variables and ψ is a countable conjunction of formulas each of which is computable Σ_β or computable Π_β for some $\beta < \alpha$,*
  - $\varphi(\overline{x})$ *is* computable Π_α *if it is a c.e. conjunction of formulas $(\forall\overline{u})\,\psi(\overline{u},\overline{x})$, where $\overline{u}$ is a countable tuple of variables and ψ is a countable disjunction of formulas each of which is computable Σ_β or computable Π_β for some $\beta < \alpha$.*

We consider structures $\mathcal{A}$ with universe a subset of ω_1. As in the standard setting, we identify a structure $\mathcal{A}$ with its atomic diagram $D(\mathcal{A})$, where this is a subset of L_{ω_1}. As we have said above, Gödel's function lets us identify elements of L_{ω_1} with elements of ω_1. In Proposition 3.1 below, when we say that a relation is Σ^0_α or Π^0_α relative to $\mathcal{A}$, we are using the first definition of the arithmetical hierarchy, relativized to $D(\mathcal{A})$. The computable infinitary formulas are as above.

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

Proof. The proof is by induction on α. First, let $\alpha = 0$. The formula $\varphi(\overline{x})$ is computable Σ_0 and computable Π_0 if it is a quantifier-free formula of $L_{\omega_1,\omega}$. Satisfaction of such formulas by countable tuples in $\mathcal{A}$ is computable relative to $D(\mathcal{A})$. Next, let $\alpha > 0$ and suppose $\varphi(\overline{x})$ is computable Σ_α, a c.e. disjunction of formulas $(\exists\overline{u})\,\psi(\overline{x},\overline{u})$, where ψ is a countable conjunction of formulas each of which is computable Π_β for some $\beta < \alpha$. Using $\Delta^0_\alpha(D(\overline{A}))$, we can determine whether a given tuple $(\overline{a},\overline{b})$ satisfies the conjuncts of such a $\psi(\overline{x},\overline{u})$. For a given $\overline{a}$, we can search for a disjunct and a tuple $\overline{b}$ witnessing that $\varphi(\overline{a},\overline{b})$ holds in $\mathcal{A}$. This shows that the relation defined by $\varphi(\overline{x})$ is Σ^0_α relative to $\mathcal{A}$. The case where $\varphi(\overline{x})$ is computable Π_α is dual.

□

When we prove the converse of Proposition 3.1, we shall also use computable propositional formulas.

Definition 13 (Computable infinitary propositional formulas)**.** *Let P be a computable set of propositional variables.*

- *φ is* computable Σ_0 *and* computable Π_0 *if it is a formula of the propositional language P_{ω_1} (allowing countable disjunctions and conjunctions).*
- *For $\alpha > 0$,*
  - *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 $\beta < \alpha$.*
  - *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 $\beta < \alpha$.*

A *structure* for the propositional language P is a set $S \subseteq P$. We have the analogue of Proposition 3.1. Truth of computable Σ_α formulas in S is Σ^0_α relative to S, and truth of computable Π_α formulas in S is Π^0_α relative to S.

3.2 Second definition

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.

Definition 14 (Computable infinitary predicate formulas)**.** *Let L be a computable predicate language, as above.*

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

- *For* $\alpha > 0$,
  - $\varphi(\overline{x})$ *is* computable Σ_α *if it is a c.e. disjunction of formulas* $(\exists \overline{u})\,\psi(\overline{u},\overline{x})$, *where* $\overline{u}$ *is a countable tuple of variables and* ψ *is computable* Π_β *for some* $\beta < \alpha$.
  - $\varphi(\overline{x})$ *is* computable Π_α *if it is a c.e. conjunction of formulas* $(\forall \overline{u})\,\psi(\overline{u},\overline{x})$, *where* $\overline{u}$ *is a countable tuple of variables and* ψ *is computable* Σ_β *for some* $\beta < \alpha$.

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

Proposition 3.2. *Let* $\mathcal{A}$ *be an* L*-structure. If the formula* $\varphi(\overline{x})$ *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* $\mathcal{A}$*, uniformly.*

Proof. The proof is by induction on α. For $\alpha = 0$, there is no difference between the two sets of definitions. Let $\alpha > 0$ and suppose $\varphi(\overline{x})$ is computable Σ_α, a c.e. disjunction of formulas $(\exists \overline{u})\,\psi(\overline{x},\overline{u})$, where ψ is computable Π_β for some $\beta < \alpha$. We must show that the relation R defined by φ is Σ^0_α relative to $\mathcal{A}$. For each $\psi(\overline{x},\overline{u})$, and each countable ordinal γ, let $R_{\psi,\gamma}$ consist of the tuples $\overline{a}$ such that $(\exists \overline{b} \in L_\gamma)\mathcal{A} \models \psi(\overline{a},\overline{b})$. The relation $R_{\psi,\gamma}$ is Π^0_β relative to $\mathcal{A}$. The relation R is the c.e. union of these. A dual argument shows that the relation defined by a computable Π_α formula is Π^0_α relative to $\mathcal{A}$.

□

Definition 15 (Computable infinitary propositional formulas)**.** *Let* P *be a computable propositional language, as above.*

- *A formula* φ *is* computable Σ_0 *and* computable Π_0 *if it is a formula of* P_{ω_1}.
- *For* $\alpha > 0$,
  - φ *is* computable Σ_α *if it is a c.e. disjunction of formulas each of which is computable* Π_β *for some* $\beta < \alpha$.
  - φ *is* computable Π_α *if it is a c.e. conjunction of formulas each of which is computable* Σ_β *for some* $\beta < \alpha$.

For both sets of definitions, our computable infinitary formulas are in "normal form". Given a formula φ, we write $neg(\varphi)$ for the dual formula that is logically equivalent to the negation. It is easy to see that if φ is computable Σ_α, then $neg(\varphi)$ is computable Π_α, and vice versa.

Remark. As above, for $S \subseteq P$, truth of computable Σ_α (or computable Π_α) formulas in S is Σ^0_α (Π^0_α) relative to S, uniformly.

4 Relatively intrinsically arithmetical relations

Recall that for a computable language L, a "computable L-structure, $\mathcal{A}$" is a structure such that the set of codes for sentences in the atomic diagram of $\mathcal{A}$ is computable. We define what it means for a relation to be relative intrinsically Σ^0_α on $\mathcal{A}$. The definition is the same as in the standard setting, except that the terms "computable" and "Σ^0_α relative to" are understood in the new sense. It should also be noted that our definition is actually two definitions according to the two different notions of "Σ^0_α."

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, but they look the same.

Theorem 4.1. *Let $1 \leq \alpha < \omega_1$. Let $\mathcal{A}$ be a computable structure, and let R be a relation on $\mathcal{A}$. Then the following are equivalent:*

1. *R is relatively intrinsically Σ^0_α on $\mathcal{A}$,*
2. *R is defined in $\mathcal{A}$ by a computable Σ_α formula $\varphi(\overline{c}, \overline{x})$, with a countable tuple of parameters $\overline{c}$.*

For simplicity, we suppose that $\mathcal{A}$ has universe equal to ω_1, and that R is unary. 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 [2] and [3]. We build a generic copy $\mathcal{B}$ of $\mathcal{A}$ by building 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 will write $S_{(\Sigma,\beta.\gamma)}(\mathcal{B})$ for the set $W_\gamma^{\Delta^0_\beta(\mathcal{B})}$. We write $S_{(\Pi,\beta.\gamma)}(\mathcal{B})$ for the complement. We identify the structure $\mathcal{B}$, under construction, with its atomic diagram. For our forcing language, we need formulas with the meanings below.

- $b \in \mathcal{B}$,
- $b \notin \mathcal{B}$,
- $b \in \Delta^0_\beta(\mathcal{B})$,
- $b \notin \Delta^0_\beta(\mathcal{B})$,
- $b \in S_{(\Sigma,\beta.\gamma)}(\mathcal{B})$,

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

We use a propositional language in which the propositional variables are the atomic sentences involving symbols from L, R, and constants from ω_1. We will identify propositional variables with their codes. The set of codes for propositional variables is a computable set. We write $neg(\varphi)$ for a formula in normal form that is logically equivalent to $\neg\varphi$. We switch disjunctions with conjunctions and we replace a propositional variable by its negation, and vice versa.

- To say that $b \in \mathcal{B}$: if b is a propositional variable or the negation of one, we write b, and if it is not a propositional variable or the negation of one, we write $\bot$.
- To say that $b \notin \mathcal{B}$, if b is a propositional variable, we write $\neg b$, and if $b = \neg c$, where c is a propositional variable, we write c. If b is neither a propositional variable nor the negation of one, then we write $\top$.

Recall that $\Delta^0_1(\mathcal{B})$ is just (the atomic diagram of) $\mathcal{B}$. So, we have formulas saying that $b \in \Delta^0_1(\mathcal{B})$ and $b \notin \Delta^0_1(\mathcal{B})$.

- Suppose β is a successor ordinal—$\beta = \delta + 1$. Then $\Delta^0_\beta(\mathcal{B})$ is the jump of $\Delta^0_\delta(\mathcal{B})$. To say that $b \in \Delta^0_\beta(\mathcal{B})$, if b is a pair (γ, c), we recall the set U from Proposition 1.3, and we take the disjunction over $\rho \in 2^{<\omega_1}$ such that $(\rho, \gamma, c) \in U$ of formulas saying that $x \in \Delta^0_\delta(\mathcal{B})$ if $\rho(x) = 1$ and $x \notin \Delta^0_\delta(\mathcal{B})$ if $\rho(x) = 0$. If b is not a pair (γ, c), we write $\bot$. To say that $b \notin \Delta^0_\beta(\mathcal{B})$, we apply neg to the formula above.
- Suppose β is a limit ordinal. To say that $c \in \Delta^0_\beta(\mathcal{B})$, if c is a pair (δ, d), where $\delta < \beta$, we take the formula saying that $d \in \Delta^0_\delta(\mathcal{B})$, and if c is not such a pair, then we write $\bot$. To say that $c \notin \Delta^0_\beta(\mathcal{B})$, if c is a pair (δ, d), where $\delta < \beta$, then we take the formula saying that $d \notin \Delta^0_\delta(\mathcal{B})$, and if c is not such a pair, then we write $\top$.
- To say that $b \in S_{(\Sigma,\beta,\gamma)}(\mathcal{B})$, we take the disjunction over $\rho \in 2^{<\omega_1}$ such that the triple $(\rho, \gamma, b) \in U$ of formulas saying that $c \in \Delta^0_\beta(\mathcal{B})$ if $\rho(c) = 1$ and $c \notin \Delta^0_\beta(\mathcal{B})$ if $\rho(c) = 0$.
- To say that $b \in S_{(\Pi,\beta,\gamma)}(\mathcal{B})$, we apply neg to the formula saying that $b \in S_{(\Sigma,n+1,\gamma)}$.
- To say that $R' = S_{(\Sigma,\beta,\gamma)}(\mathcal{B})$, we take the conjunction over all b of the formulas $\bigwedge\!\!\!\bigwedge_b (b \in S_{(\Sigma,\beta,\gamma)}(\mathcal{B}) \leftrightarrow R'b)$.

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_γ saying that R' is equal to the set with index (Σ, α, γ) relative to $\mathcal{B}$, and the $\Sigma_{\alpha+1}$ formulas $neg(\chi_\gamma)$. We define forcing for the formulas in T.

Definition 17 (Definition of forcing). *Let p be a forcing condition. We define the relation $p \Vdash \varphi$, for φ in our propositional language.*

- *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 the structure $(\mathcal{A}, R)$.*
- *Suppose φ is computable Σ_β, for $\beta \geq 1$, say φ is the c.e. disjunction of formulas ψ_i, where ψ_i is a countable conjunction of formulas $\psi_{i,j}$ and for each j, $\psi_{i,j}$ is Σ_γ or Π_γ for some $\gamma < \beta$. Then $p \Vdash \varphi$ if there is some i such that for all j, $p \Vdash \psi_{i,j}$.*
- *Suppose φ is computable Π_β, for $\beta \geq 1$, say φ is the c.e. conjunction of formulas ψ_i, where ψ_i is a countable disjunction of formulas $\psi_{i,j}$ and for each j, $\psi_{i,j}$ is computable Π_γ or computable Σ_γ for some $\gamma < \beta$. Then $p \Vdash \varphi$ if for all i and all $q \supseteq p$, there exist $r \supseteq q$ and j such that $r \Vdash \psi_{i.j}$.*

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 $q \Vdash neg(\varphi)$.*

From the definition above, forcing of computable Π_β formulas does not appear to be Π^0_β. The lemma below gives an alternative condition, which is Π^0_β.

Lemma 4.5. *Let φ be a computable Π_β formula, the c.e. intersection of formulas ψ_i, where ψ_i is a countable disjunction of formulas $\psi_{i,j}$, for $j \in \omega$, each computable Π_γ or computable Σ_γ for some $\gamma < \beta$. For a forcing condition p, $p \Vdash \varphi$ iff for all i and all $q \supseteq p$, it is not the case that for all $j \in \omega$, $q \Vdash \psi_{i,j}$.*

Proof. We show that for all i, the following are equivalent.

1. for all $q \supseteq p$, there exist $r \supseteq q$ and j such that $r \Vdash \psi_{i,j}$,
2. for all $q \supseteq p$, it is not the case that $(\forall j)\, q \Vdash neg(\psi_{i,j})$.

First suppose (1). If (2) fails, we would have $q \supseteq p$ such that $(\forall j)\, q \Vdash neg(\psi_{i,j})$. By the Extension and Consistency lemmas, we cannot have $r \supseteq q$ and $j \in \omega$ such that $r \Vdash \psi_{i,j}$. This contradicts (1). Therefore, (2) holds. Now, suppose (2). If (1) fails, we would have $q \supseteq p$ such that there do not exist $r \supseteq q$ and j with $r \Vdash \psi_{i,j}$. We build a chain $(r_j)_{j \in \omega}$ of extensions of q, where $r_0 \supseteq q$ forces $neg(\psi_{i,0})$ and $r_{j+1} \supseteq r_j$ forces $neg(\psi_{i,j+1})$. Let $r = \cup_j r_j$. Then $r \supseteq p$ and for all j, $r \Vdash neg(\psi_{i,j})$, contradicting (2).

□

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

1. *if* $\delta < \delta'$*, then* $p_\delta \supseteq p_{\delta'}$*,*

2. *for all* $\varphi \in T$*, there is some* δ *such that* p_δ *decides* φ*,*

3. *for all* $a \in \omega_1$*, there is some* δ *such that* $a \in ran(p_\delta)$*.*

It follows from the lemmas that we can form a complete forcing sequence. For limit δ, we let $p_\delta = \cup_{\delta'<\delta} p_{\delta'}$. Let $F = \cup_\delta p_\delta$ for $\delta < \omega_1$. 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 denote in the same way.

Lemma 4.6 (Truth and Forcing Lemma)**.** *For* $\varphi \in T$*,* $(\mathcal{B}, R') \models \varphi$ *iff there is some* δ *such that* $p_\delta \Vdash \varphi$*.*

Proof.

1. Suppose φ is computable Σ_0 and computable Π_0. Take δ such that $dom(p_\delta)$ includes all of the constants that appear in the propositional variables in φ. If p_δ makes φ true in $(\mathcal{A}, R)$, then $p_\delta \Vdash \varphi$ and $(\mathcal{B}, R') \models \varphi$. If p_δ makes φ false in $(\mathcal{A}, R)$, then $p_\delta \Vdash neg(\varphi)$ and $(\mathcal{B}, R') \models neg(\varphi)$.

2. Suppose φ is computable Σ_β, the c.e. disjunction of formulas ψ_i, where ψ_i is a countable conjunction of formulas $\psi_{i,j}$, and for each j, $\psi_{i,j}$ is computable Π_γ or computable Σ_γ for some $\gamma < \beta$. First, suppose $(\mathcal{B}, R') \models \varphi$, then there is some i such that for all j, $(\mathcal{B}, R') \models \psi_{i,j}$. By the induction hypothesis, for each j, there is some $\delta_{i,j}$ such that $p_{\delta_{i,j}} \Vdash \psi_{i,j}$. Let δ be the sup of the $\delta_{i,j}$. Then $p_\delta \Vdash \psi_{i,j}$ for all j, so $p_\delta \Vdash \varphi$. Now, suppose $p_\delta \Vdash \varphi$. By the definition of forcing, there is some i such that for all j, $p_\delta \Vdash \psi_{i,j}$. By the induction hypothesis, for all j, $\psi_{i,j}$ is true, so ψ_i is true and so is φ.

3. Suppose φ is computable Π_β, the c.e. conjunction of formulas ψ_i, where ψ_i is a countable disjunction of formulas $\psi_{i,j}$, each computable Σ_γ or computable Π_γ for some $\gamma < \beta$. First, suppose $(\mathcal{B}, R') \models \varphi$. This means that for all i, ψ_i is true. For some δ, p_δ forces either φ or $neg(\varphi)$. We show that p_δ cannot force $neg(\varphi)$. If $p_\delta \Vdash neg(\varphi)$, then for some i, for all j, $p_\delta \Vdash neg(\psi_{i,j})$. By the induction hypothesis, for all j, $neg(\psi_{i,j})$ is true, so the conjunction is true, and this is $neg(\psi_i)$, contradicting the fact that ψ_i is true. Now, suppose that for some δ, $p_\delta \Vdash \varphi$. For each i and each $q \supseteq p_\delta$, there is some $r \supseteq q$ such that for some j, $r \Vdash \psi_{i,j}$. If φ is false, then for some i, ψ_i is false, which means that for all j, $\psi_{i,j}$ is false. By the induction hypothesis, for each of the countably many j, there is some p_{δ_j} forcing $neg(\psi_{i,j})$. Let q be the union of p_δ and the p_{δ_j}. Since $q \supseteq p_\delta$, there is some $r \supseteq q$ such that for some j, $r \Vdash \psi_{i,j}$. Then r forces both $\psi_{i,j}$ and $neg(\psi_{i,j})$, a contradiction. Therefore, φ must be true.

□

Let T' be the set of formulas in T that do not involve R. We shall prove that forcing for these formulas is definable in $\mathcal{A}$.

Lemma 4.7 (Definability of forcing)**.** *For any $\varphi \in T'$, and for any tuples $\overline{b}$ 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 computable Π_β, for $1 \le \beta$, then $Force_{\overline{b},\varphi}(\overline{x})$ is also computable Σ_β, or computable Π_β.*

Proof. We suppose that the elements of $\overline{b}$ are distinct, and the variables in $\overline{x}$ are distinct. We have a simple formula $force_{\overline{b}}(\overline{x})$ saying that the correspondence taking b_i to x_i is a forcing condition—take the conjunction of formulas $x_i \neq x_j$, where $i \neq j$. Now, we give the formulas $Force_{\overline{b},\varphi}(\overline{x})$ by induction on φ.

1. Suppose φ is computable Σ_0 and computable Π_0. While φ is really propositional, it is convenient to think of it also as a quantifier-free predicate sentence involving predicate symbols from the language of $\mathcal{A}$ and constants from the universe of $\mathcal{B}$. Suppose that the constants that appear in φ are all in $\overline{b}$, and let φ' be the result of replacing each occurrence of b_i in φ by x_i. Then $Force_{\overline{b},\varphi}(\overline{x}) = (force_{\overline{b}}(\overline{x})\ \&\ \varphi')$. If the constants that appear in φ are not all in $\overline{b}$, then $Force_{\overline{b},\varphi}(\overline{x}) = \bot$. In either case, the formula $Force_{\overline{b},\varphi}(\overline{x})$ is computable Σ_0 and computable Π_0.

2. Suppose φ is computable Σ_β, the c.e. disjunction of formulas ψ_i, where ψ_i is a countable conjunction of formulas $\psi_{i,j}$, each of which is computable Π_γ or computable Σ_γ for some $\gamma < \beta$. Let $Force_{\overline{b},\varphi}(\overline{x})$ be $\bigvee\!\!\!\bigvee_i \bigwedge\!\!\!\bigwedge_j Force_{\overline{b},\psi_{i,j}}(\overline{x})$. Each $Force_{\overline{b},\psi_{i,j}}(\overline{x})$ is computable Π_γ or computable Σ_γ for some $\gamma < \beta$. Then $Force_{\overline{b},\varphi}(\overline{x})$ is computable Σ_α.

3. Suppose that φ is computable Π_β, the c.e. conjunction of formulas ψ_i, where ψ_i is a countable disjunction of formulas $\psi_{i,j}$, each of which is computable Σ_γ or computable Π_γ for some $\gamma < \beta$. We think of the alternative definition of forcing for such formulas. We let $Force_{\overline{b},\varphi}(\overline{x})$ be a computable Π_β formula saying that for all i and for all $\overline{d}$ and $\overline{u}$, if the correspondence taking $\overline{b}, \overline{d}$ to $\overline{x}, \overline{u}$ is a forcing condition q, then it is not the case that for all $j \in \omega$, $q \Vdash neg(\psi_{i,j})$. We write

$$\bigwedge\!\!\!\bigwedge_i \bigwedge\!\!\!\bigwedge_{\overline{d}} (\forall \overline{u})\,(force_{\overline{b},\overline{d}}(\overline{x},\overline{u}) \to \bigwedge\!\!\!\bigwedge_j neg(Force_{\overline{b},\overline{d},neg(\psi_{i,j})}(\overline{x},\overline{b}))$$

This is equivalent to a c.e. conjunction of countable conjunctions of computable Σ_γ and computable Π_γ formulas for $\gamma < \beta$.

□

We are ready to complete the proof. By assumption and Lemma 4.5, some p forces $S_{(\Sigma,\alpha,\gamma)}(\mathcal{B}) = R'$, for some γ. Say that 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 $q \Vdash b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$. For any b, the formula saying that $b \in S_{(\Sigma,\alpha,\gamma)}$ is computable Σ_α. We can write 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 c.e. 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 computable Σ_α.

□

We have proved the first version of the theorem, using the first set of definitions. We now consider the second version of the theorem, 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. The outline of the proof is the same as above. Recall that $S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$ is the c.e. union of the sets $S_{(\Pi,\beta,\delta)}(\mathcal{B})$ such that $(\Pi, \beta, \delta) \in W_\gamma$ and $\beta < \alpha$. 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 $\alpha = 1$, $S_{(\Sigma,1,\gamma)}(\mathcal{B}) = W_\gamma^{\mathcal{B}}$, and $S_{(\Pi,1,\gamma)}(\mathcal{B})$ is the complementary set. For our forcing language, we need formulas with the meanings below.

- $b \in \mathcal{B}$,
- $b \notin \mathcal{B}$,
- $b \in S_{(\Sigma,1,\gamma)}(\mathcal{B})$,
- $b \in S_{(\Pi,1,\gamma)}$, or $b \notin S_{(\Sigma,1,\gamma)}(\mathcal{B})$,
- $b \in S_{(\Sigma,\beta,\gamma)}(\mathcal{B})$, or $b \in S_{(\Pi,\beta,\gamma)}(\mathcal{B})$,
- $R' = S_{(\Sigma,\beta,\gamma)}(\mathcal{B})$.

Again, the propositional variables are the atomic sentences involving symbols from L, R, and constants from ω_1.

- The formulas saying that $b \in \mathcal{B}$ and $b \notin \mathcal{B}$ are the same as before. To say that $b \in \mathcal{B}$, we write b if b is a propositional variable, $\neg c$ if b is the negation of a propositional variable c, and $\perp$ otherwise. To say that $b \notin \mathcal{B}$, we write $\neg b$ if b is a propositional variable, c if b is the negation of c, and $\top$ if b is neither a propositional variable nor the negation of one.
- To say that $b \in S_{(\Sigma,1,\gamma)}(\mathcal{B})$, we take the disjunction, over $\rho \in 2^{<\omega}$ such that (ρ, γ, b) is in the relation U, of the conjunction of formulas saying $x \in \mathcal{B}$, for $\rho(x) = 1$, and formulas saying $x \notin \mathcal{B}$, for $\rho(b) = 0$. To say that $b \in S_{(\Pi,1,\gamma)}(\mathcal{B})$, or $b \notin S_{(\Sigma,1,\gamma)}(\mathcal{B})$, we apply *neg* to the formula saying $b \in S_{(\Sigma,1,\gamma)}(\mathcal{B})$.

- For $\delta > 1$, to say that $b \in S_{(\Sigma,\delta,\gamma)}(\mathcal{B})$, we take the disjunction, over $(\Pi, \delta', \gamma') \in W_\gamma$ with $1 \leq \delta' < \delta$, of formulas saying that $b \in S_{(\Pi,\delta',\gamma')}(\mathcal{B})$. To say that $b \in S_{(\Pi,\delta,\gamma)}(\mathcal{B})$, we apply *neg* to the formula saying that $b \in S_{(\Sigma,\delta,\gamma)}(\mathcal{B})$.

- To say that $R' = S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$, we take the formula saying $\bigwedge\!\!\!\bigwedge_b (b \in S_{(\Sigma,\alpha,\gamma)}(\mathcal{B}) \leftrightarrow R'b)$.

We let T include the computable Σ_β and Π_β formulas, for countable ordinals $\beta \leq \alpha$, plus the formula saying $R' = S_{(\Sigma,\alpha,\gamma)}(\mathcal{B})$ and the result of applying *neg* to this formula.

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

- *Suppose φ is computable Σ_0 and Π_0. We say p forces φ, or $p \Vdash \varphi$, 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_β, for $\beta \geq 1$, a c.e. disjunction of formulas ψ_i, where each ψ_i is computable Π_γ for some $\gamma < \beta$. Then $p \Vdash \varphi$ if $p \Vdash \psi_i$, for some i.*

- *Suppose φ is computable Π_β, for $\beta \geq 1$, a c.e. conjunction of formulas ψ_i, where each ψ_i is computable Σ_γ for some $\gamma < \beta$. 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 4.8 (Extension)**.** *If $p \Vdash \varphi$ and $q \supseteq p$, then $q \Vdash \varphi$.*

Lemma 4.9 (Consistency)**.** *It is not the case that $p \Vdash \varphi$ and $p \Vdash neg(\varphi)$.*

Lemma 4.10 (Density)**.** *For all p and φ, there exists $q \supseteq p$ such that $q \Vdash \varphi$ or $q \Vdash neg(\varphi)$.*

As for the first definition, we could show that if φ is computable Π_β, a c.e. conjunction of formulas ψ_i, each computable Σ_γ for some $\gamma < \beta$, then $p \Vdash \varphi$ iff for all i and all $q \supseteq p$, q does not force $neg(\psi_i)$. This is not necessary, since for the second definition, the relation $p \Vdash \varphi$ is easily seen to be Π^0_β.

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

Lemma 4.11 (Truth and forcing Lemma)**.** *For $\varphi \in T$, $(\mathcal{B}, R') \models \varphi$ iff there is some β such that $p_\beta \Vdash \varphi$.*

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

Lemma 4.12 (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 $1 \le \beta \le \alpha$, then $Force_{\overline{b},\varphi}(\overline{x})$ is also computable Σ_β, or computable Π_β.*

Proof. First, suppose φ is computable Σ_0 and computable Π_0. If $\overline{b}$ includes all of the constants that appear in the propositional variables of φ, then we let $Force_{\overline{b},\varphi}(\overline{x})$ be $force_{\overline{b}}(\overline{x})$ & φ', where φ' is the result of replacing the occurrences of the constant b_i in φ by the corresponding variable x_i. If $\overline{b}$ does not include all of the constants of φ, then $Force_{\overline{b},\varphi}(\overline{x})$ is $\perp$. Suppose φ is computable Σ_β, a c.e. disjunction of formulas ψ_i, each of which is computable Π_γ for some $\gamma < \beta$. We let $Force_{\overline{b},\varphi}(\overline{x})$ be

$$\bigvee\!\!\!\bigvee_i (force_{\overline{b}}(\overline{x}) \ \&\ Force_{\overline{b},\psi_i}(\overline{x}))$$

This is a computable Σ_β predicate formula. Finally, suppose φ is computable Π_β, a c.e. conjunction of formulas ψ_i, each of which is computable Σ_γ for some $\gamma < \beta$. We let $Force_{\overline{b},\varphi}(\overline{x})$ be the conjunction of $force_{\overline{b}}(\overline{x})$ and the following formulas, one for each i and $\overline{c}$:

$$(\forall \overline{u})\,[force_{\overline{b},\overline{c}}(\overline{x},\overline{u}) \to \bigvee\!\!\!\bigvee_{\overline{d}} ((\exists \overline{v})(force_{\overline{b},\overline{c},\overline{d}}(\overline{x},\overline{u},\overline{v}) \ \&\ Force_{\overline{b},\overline{c},\overline{d},\psi_i}(\overline{x},\overline{u},\overline{v}))]$$

This is logically equivalent to a computable Π_β predicate formula.

□

We are ready to complete the proof of the second version 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 c.e. 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.

□

5 Which definition is better?

The two definitions of the arithmetical hierarchy in the setting of ω_1 are not equivalent. Each definition yields a result saying that a relation is relatively intrinsically Σ^0_α on $\mathcal{A}$ iff it is defined by a computable Σ_α formula. So, we do not have evidence that one definition is more productive. We would like to claim that the first definition is better. In the standard setting, an element enters a Σ^0_5 set based on finitely much Δ^0_5 information. We are using the full power of the oracle. In our first definition, an element enters a Σ^0_ω set based on countably many pieces of Δ^0_ω information. We may use the full power of the Δ^0_ω oracle. In our second definition, an element enters a Σ^0_ω set by entering one of a c.e. family of sets, each of which is Π^0_n for some $n < \omega$. We never use the full power of Δ^0_ω. Using this reasoning, it seems more natural to use the first definition for the arithmetical hierarchy. We are grateful to Joe Mileti for helpful discussions of this point. We are also grateful to Sy Friedman, for telling us about some related work of Jensen, on master codes, in which he also has a choice of approaches, and makes a choice like ours ([5], Section 2).

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] R. Jensen, "The fine structure of the constructible hierarchy", *Annals of Math. Logic*, vol. 4 (1972), pp. 229308.

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