2013

The arithmetical hierarchy in the setting of $\omega_1$

Jacob Carson
Jesse Johnson
Julia F. Knight
Karen Lange
klange2@wellesley.edu
Charles McCoy CSC

See next page for additional authors

Follow this and additional works at: http://repository.wellesley.edu/scholarship

Recommended Citation
The arithmetical hierarchy in the setting of $\omega_1$ computability, J. Carson, J. Johnson, J. Knight, K. Lange, C. McCoy, and J. Wallbaum, Computability, 2(2) (2013), 93-105.

This Article is brought to you for free and open access by Wellesley College Digital Scholarship and Archive. It has been accepted for inclusion in Faculty Research and Scholarship by an authorized administrator of Wellesley College Digital Scholarship and Archive. For more information, please contact ir@wellesley.edu.
The arithmetical hierarchy in the setting of $\omega_1$

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 $\omega_1$, 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 $\Sigma^0_\alpha$ just in case it is definable by a computable $\Sigma^0_\alpha$ 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 $\omega_1$. 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 $L_{\omega_1}$. In fact, every subset of $\omega_1$ is “amenable” for $L_{\omega_1}$; 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 $\Sigma^0_\alpha$ and $\Pi^0_\alpha$ subsets of $\omega_1$, for countable ordinals $\alpha$. We do this in two different ways. The first is based on the standard definition of the hyperarithmetical hierarchy—a set is $\Sigma^0_\alpha$ if it is c.e. relative to a complete $\Delta^0_\alpha$ oracle. The second definition is based on the definition of the effective Borel hierarchy—a set is $\Sigma^0_\alpha$ if it is a c.e. union of sets each of which is $\Pi^0_\beta$ for some $\beta \lt \alpha$.

In Section 3, we define computable infinitary $\Sigma_\alpha$ 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 $L_{\omega_2, \omega_1}$. 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 $\Sigma_\alpha$ if and
only if it is defined in $A$ by a computable $\Sigma_\alpha$ 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 $L_{\omega_1}$ 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 $\Delta_0$, 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 $L_{\omega_1}$.

- A relation $R \subseteq (L_{\omega_1})^n$ is computably enumerable, or c.e., if it is defined in $L_{\omega_1}$ by a $\Sigma_1$-formula $\varphi(c, x)$, 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 $\Sigma_1$ definition for a function $f$ says that there exists a sequence of steps leading to the value of $f$ at a given ordinal $\alpha$. 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 $L_{\omega_1}$ such that the relation $g(\alpha) \in g(\beta)$ is computable. The function $g$ provides ordinal codes for sets—$\alpha$ is the code for $g(\alpha)$. There is also a computable function $\ell$ taking $\alpha$ to the code for $L_\alpha$. From this, we see that computing in $\omega_1$ is essentially the same as computing in $L_{\omega_1}$. 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 $\alpha$, 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 $\Sigma_1$-definable—the set of ordinal codes for sequences in $R$ is c.e.

- A relation of arity $\alpha$ is computable if it is both c.e. and co-c.e.
A function \( f : (L_{\omega_1})^\alpha \rightarrow L_{\omega_1} \) is computable if its graph \( \{ \langle \pi, f(\pi) : \pi \in \text{dom} f \rangle \} \) is a c.e. relation.

We have a c.e. set \( C \) of codes for pairs \( (\varphi, \tau) \), representing \( \Sigma_1 \) definitions, where \( \varphi(\pi, \tau) \) is a \( \Sigma_1 \)-formula and \( \tau \) is a tuple of parameters appropriate for \( \pi \). Note that \( \pi \) and \( \tau \) can be countable tuples.

We have a computable function \( h \) mapping \( \omega_1 \) onto \( C \).

**Definition 3.** For \( \alpha < \omega_1 \), \( \alpha \) is a c.e. index for \( X \) if \( h(\alpha) \) is the code for a pair \( (\varphi, \tau) \), where \( \varphi(\tau, x) \) is a \( \Sigma_1 \) definition of \( X \) in \( (L_{\omega_1}, \epsilon) \).

**Notation:** We write \( W_\alpha \) for the c.e. set with index \( \alpha \).

Suppose \( W_\alpha \) is determined by the pair \( (\varphi, \tau) \); i.e., \( \varphi(\tau, x) \) is a \( \Sigma_1 \) definition.

**Definition 4.** We say that \( x \) is in \( W_\alpha, \beta \) at stage \( \beta \), and we write \( x \in W_\alpha, \beta \), if \( L_\beta \) contains \( x \), the parameters \( \tau \), and witnesses making the formula \( \varphi(\tau, x) \) true.

**Remark.** The relation \( x \in W_\alpha, \beta \) is computable. After all, in the current setting, countable sets, such as \( L_\beta \), 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 \( \Gamma \) 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 \( \Sigma_1 \) definable in \( L_{\omega_1} \).

\[ \square \]

**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 \((\alpha, \beta)\) 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 \( \omega_1 \), 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 \( \Sigma_1 \) definition for \( K \), saying that there exists \( \beta \) such that \( L_\beta \) contains the pair \( h(\alpha) = (\varphi(\pi, y), \bar{\pi}) \), and \( L_\beta = \varphi(\pi, y) \). 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 \( \Sigma_1 \)-definable in \((L_{\omega_1}, \varepsilon, X)\).
- 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 $\alpha$ such that $h(\alpha) = (\varphi, c)$, where $\varphi$ is a $\Sigma_1$ formula (in the language with $\varepsilon$ and a predicate symbol for $X$), and $\varphi(\varepsilon, x)$ defines $R$ in \((L_{\omega_1}, \varepsilon, X)\).

Notation: We write $W^X_\alpha$ for the c.e. set with index $\alpha$ relative to $X$.

Proposition 1.3. There is a c.e. set $U$ of the codes for triples $(\sigma, \alpha, \beta)$ such that $\sigma \in 2^\rho$ (for some countable ordinal $\rho$), and for all $X$ with characteristic function extending $\sigma$, $\beta \in W^X_\alpha$.

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 $X' = \{(\alpha, y) : y \in W^X_\alpha\}$.

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 $\alpha$, $X^{(\alpha)}$ is the set of codes for pairs $(\beta, x)$ 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 $\Delta^0_n$ for $\varphi^{(n-1)}$, and for countable ordinals $\alpha \geq \omega$, we write $\Delta^0_\alpha$ for $\varphi^{(\alpha)}$. We can relativize to a set $X$. For finite $n \geq 1$, we let $\Delta^0_n(X) = X^{(n-1)}$, and for $\alpha \geq \omega$, we let $\Delta^0_\alpha(X) = X^{(\alpha)}$.

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 hyperarith- metical hierarchy in the usual setting.

Definition 9. Let $R$ be a relation on $\omega_1$.

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

We may relativize this.

Definition 10. Let $R$ be a relation on $\omega_1$.

- $R$ 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 complementary relation $\neg R$ is $\Sigma^0_\alpha(X)$.

We assign indices to the $\Sigma^0_\alpha$ and $\Pi^0_\alpha$ sets, for $\alpha \geq 1$. We ignore the case where $\alpha = 0$. The indices have the form $(\Sigma, \alpha, \gamma)$ and $(\Pi, \alpha, \gamma)$. The first two components indicate that the set is $\Sigma^0_\alpha$, or $\Pi^0_\alpha$. The set with index $(\Sigma, \alpha, \gamma)$ is $W^0_{\Delta^0_\gamma}$, and the set with index $(\Pi, \alpha, \gamma)$ is the complementary set. When we relativize to a set $X$, we use the same indices. The set with index $(\Sigma, \alpha, \gamma)$ relative to $X$ is $W^0_{\Delta^0_\gamma(X)}$, and the set with index $(\Pi, \alpha, \gamma)$ 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 $\omega_1$.

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

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

We may assign indices to the $\Sigma^0_n$ and $\Pi^0_n$ sets in a natural way. We ignore $\alpha = 0$. For $\alpha = 1$, $(\Sigma, 1, \gamma)$ is the index for the c.e. set $W_\gamma$, and $(\Pi, 1, \gamma)$ is the index for the complementary set. For $\alpha > 1$, $(\Sigma, \alpha, \gamma)$ is the union of the sets with indices in $W_\gamma$ of the form $(\Pi, \beta, \delta)$, where $1 \leq \beta < \alpha$. Similarly, $(\Pi, \alpha, \gamma)$ is the index for the intersection of the sets with indices in $W_\gamma$ of the form $(\Sigma, \beta, \delta)$, 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 $W_\gamma^X$; the set with index $(\Pi, 1, \gamma)$ relative to $X$ is the complement. For $\alpha > 1$, the set with index $(\Sigma, \alpha, \gamma)$ relative to $X$ is the union of the sets with indices (relative to $X$) in $W_\gamma$ of the form $(\Pi, \beta, \delta)$, where $1 \leq \beta < \alpha$; the set with index $(\Pi, \alpha, \gamma)$ relative to $X$ is the intersection of the sets with indices (relative to $X$) in $W_\gamma$ of the form $(\Sigma, \beta, \delta)$, where $1 \leq \beta < \alpha$.

2.3 Comparing the two definitions

We write $\Sigma^0_n(I)$, $\Pi^0_n(I)$ for the first definition, and $\Sigma^0_n(II)$, $\Pi^0_n(II)$ 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 $\Sigma^0_n$ and $\Pi^0_n$ sets and relations are the computable ones. Also, under both definitions, the $\Pi^0_n$ sets are the complements of the $\Sigma^0_n$ sets. A set is $\Sigma^0_n(I)$ iff it is c.e. relative to $\varnothing$, and a set is $\Sigma^0_n(II)$ 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 $\gamma_1$, with only bounded quantifiers, such that $y \in \varnothing'$ iff the formula $(\exists \pi)\gamma_1(\pi, y)$ holds in $L_{\omega_1}$. (We ignore the parameters.) For an ordinal $\beta$, let $Y_{1, \beta}$ be the set of $y \in L_\beta$ such that $(\exists \pi)\gamma_1(\pi, y)$ holds in $L_\beta$; i.e., we can take $\pi \in L_\beta$. The relation $Y = Y_{1, \beta}$ (on $Y$ and $\beta$) is computable. We say that $\beta$ 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 $(\exists \pi)\gamma_1(\pi, y)$ holds in $L_\beta$. The set of $\beta$ that are $1$-good is $\Pi^0_1$.

Similarly, $\varnothing^{(n+1)}$ is c.e. relative to $\varnothing^n$, so we have a formula $\gamma_{n+1}$, with only bounded quantifiers, such that $y \in \varnothing^{(n+1)}$ iff the formula $(\exists \pi)\gamma(\pi, x)$ 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 $(\exists \pi)\gamma_{n+1}(\pi, y)$ holds in $(L_\beta, Y_{n, \beta})$. The relation $Y = Y_{n+1, \beta}$ is computable. We say that $\beta$ 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 $(\exists \pi)\gamma_{n+1}(\pi, y)$ holds in $(L_\beta, Y_{n, \beta})$. The set of $\beta$ that are $(n+1)$-good is $\Pi^0_{n+1}$. 

6
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 \( \delta \), with only bounded quantifiers, such that \( x \in R \) iff the formula \( (\exists \pi)\delta(\pi, x) \) holds in \( (L_{\omega_1}, \varnothing^{(n)}) \). We have \( x \in R \) iff there is some \( \beta \) such that \( x \in L_\beta \), \( \beta \) is \( n \)-good, and \( (\exists \pi)\delta(\pi, x) \) holds in \( (L_\beta, Y_{n, \beta}) \). For each ordinal \( \alpha \), let \( R_\alpha \) be the set of \( x \) such that for some \( \beta < \alpha \), \( \beta \) is \( n \)-good, \( x \in L_\beta \), and \( (\exists \pi)\delta(\pi, x) \) holds in \( (L_\beta, Y_{n, \beta}) \). The sets \( R_\alpha \) are \( \Pi^0_n \), uniformly in \( \alpha \), and \( R \) is a c.e. union. Therefore, \( R \) is \( \Sigma^0_{n+1}(I) \).

Now, suppose \( R \) is \( \Sigma^0_{n+1}(I) \). Say \( R \) is the union of \( \Pi^0_n \) 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 \( \Pi^0_n \) set with index \( i \). The set of all pairs \((x, i)\) such that \( x \) is an element of the \( \Pi^0_n \) set with index \( i \) is \( \Pi^0_n \), 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 \( \omega \) and beyond. Under the first definition, the computation that puts a particular element into a \( \Sigma^0_\omega \) may involve \( \Delta^0_n \) information for all \( n \). Under the second definition, an element enters a \( \Sigma^0_\omega \) by entering some \( \Pi^0_n \) 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 \((\Sigma, \omega, \alpha)\)—the set is equal to the union of the sets that have indices in \( W_\alpha \) of the form \((\Pi, n, \beta)\), 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 \( \alpha \) is not in the set with second-definition index \((\Sigma, \omega, \alpha)\); we define \( S \) to diagonalize out of the class of \( \Sigma^0_\omega(II) \) sets. For each \( \alpha \) and \( n \), let \( S(\alpha, n) \) be the union of the \( \Pi^0_k \) sets with indices in \( W_\alpha \) of the form \((\Pi, k, \beta)\), with \( k < n \) for this fixed \( n \). Note that each \( S(\alpha, n) \) is \( \Sigma^0_n \). The union of these sets over all \( n \) will be the \( \Sigma^0_\omega(II) \) set with index \((\Sigma, \omega, \alpha)\). For each countable \( \alpha \), we can determine, computably relative to \( \Delta^0_\omega \), 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 \( \Delta^0_\omega(II) \), 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 \( \omega \). 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 $\Sigma_\alpha$ or $\Pi_\alpha$ for countable ordinals $\alpha$. 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 $\Sigma_\alpha$ or computable $\Pi_\alpha$ for computable ordinals $\alpha$. For more information on computable infinitary formulas in the standard setting, see [3].

In this section, we give definitions, for the setting of $\omega_1$, of the computable $\Sigma_\alpha$ and computable $\Pi_\alpha$ formulas for countable ordinals $\alpha$. Of course, c.e. disjunctions and conjunctions may be uncountable. We allow countable tuples of variables. Thus, our computable infinitary formulas are formulas of $L_{\omega_2,\omega_1}$, 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 $\Sigma_0$ and computable $\Pi_0$ 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 $\Sigma_\alpha$ 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 $\psi$ is a countable conjunction of formulas each of which is computable $\Sigma_\beta$ or computable $\Pi_\beta$ for some $\beta < \alpha$,
  - $\varphi(\overline{x})$ is computable $\Pi_\alpha$ 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 $\psi$ is a countable disjunction of formulas each of which is computable $\Sigma_\beta$ or computable $\Pi_\beta$ for some $\beta < \alpha$.

We consider structures $\mathcal{A}$ with universe a subset of $\omega_1$. As in the standard setting, we identify a structure $\mathcal{A}$ with its atomic diagram $D(\mathcal{A})$, where this is a subset of $L_{\omega_1}$. As we have said above, Gödel’s function lets us identify elements of $L_{\omega_1}$ with elements of $\omega_1$. In Proposition 3.1 below, when we say that a relation is $\Sigma^0_\alpha$ or $\Pi^0_\alpha$ 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.

8
Proposition 3.1. Let $\mathcal{A}$ be an $L$-structure. If $\varphi(\overline{x})$ is computable $\Sigma_\alpha$ (or computable $\Pi_\alpha$) $L$-formula, then the relation defined by $\varphi(\overline{x})$ in $\mathcal{A}$ is $\Sigma_0^0$ (or $\Pi_0^0$) relative to $\mathcal{A}$, uniformly.

Proof. The proof is by induction on $\alpha$. First, let $\alpha = 0$. The formula $\varphi(\overline{x})$ is computable $\Sigma_0$ and computable $\Pi_0$ 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 $\Sigma_\alpha$, a c.e. disjunction of formulas $(\exists \overline{v}) \psi(\overline{x}, \overline{v})$, where $\psi$ is a countable conjunction of formulas each of which is computable $\Pi_\beta$ for some $\beta < \alpha$. Using $\Delta_0^0(D(\mathcal{A}))$, we can determine whether a given tuple $(\overline{a}, \overline{b})$ satisfies the conjuncts of such a $\psi(\overline{x}, \overline{v})$. For a given $\overline{v}$, 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 $\Sigma_0^0$ relative to $\mathcal{A}$. The case where $\varphi(\overline{x})$ is computable $\Pi_\alpha$ 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.

- $\varphi$ is computable $\Sigma_0$ and computable $\Pi_0$ if it is a formula of the propositional language $P_{\omega_1}$ (allowing countable disjunctions and conjunctions).
- For $\alpha > 0$,
  - $\varphi$ is computable $\Sigma_\alpha$ if it is a c.e. disjunction of countable conjunctions of formulas each of which is computable $\Sigma_\beta$ or computable $\Pi_\beta$ for some $\beta < \alpha$.
  - $\varphi$ is computable $\Pi_\alpha$ if it is a c.e. conjunction of countable disjunctions of formulas each of which is computable $\Sigma_\beta$ or computable $\Pi_\beta$ 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 $\Sigma_\alpha$ formulas in $S$ is $\Sigma_0^0$ relative to $S$, and truth of computable $\Pi_\alpha$ formulas in $S$ is $\Pi_0^0$ 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 $\Sigma_0$ and computable $\Pi_0$ if it is a quantifier-free formula of $L_{\omega_1,\omega}$.
• For $\alpha > 0$,
  
  $\varphi(\overline{x})$ is computable $\Sigma_\alpha$ 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 $\psi$ is computable $\Pi_\beta$ for some $\beta < \alpha$.

  $\varphi(\overline{x})$ is computable $\Pi_\alpha$ 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 $\psi$ is computable $\Sigma_\beta$ for some $\beta < \alpha$.

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

**Proposition 3.2.** Let $A$ be an $L$-structure. If the formula $\varphi(\overline{x})$ is computable $\Sigma_\alpha$ (or computable $\Pi_\alpha$), then the relation defined by $\varphi(\overline{x})$ in $A$ is $\Sigma^0_\alpha$ (or $\Pi^0_\alpha$) relative to $A$, uniformly.

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

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

• A formula $\varphi$ is computable $\Sigma_0$ and computable $\Pi_0$ if it is a formula of $P_{\omega_1}$.

• For $\alpha > 0$,
  
  $\varphi$ is computable $\Sigma_\alpha$ if it is a c.e. disjunction of formulas each of which is computable $\Pi_\beta$ for some $\beta < \alpha$.

  $\varphi$ is computable $\Pi_\alpha$ if it is a c.e. conjunction of formulas each of which is computable $\Sigma_\beta$ for some $\beta < \alpha$.

For both sets of definitions, our computable infinitary formulas are in “normal form”. Given a formula $\varphi$, we write $\neg \varphi$ for the dual formula that is logically equivalent to the negation. It is easy to see that if $\varphi$ is computable $\Sigma_\alpha$, then $\neg \varphi$ is computable $\Pi_\alpha$, and vice versa.

**Remark.** As above, for $S \subseteq P$, truth of computable $\Sigma_\alpha$ (or computable $\Pi_\alpha$) formulas in $S$ is $\Sigma^0_\alpha$ ($\Pi^0_\alpha$) 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 \( \Sigma^0_\alpha \) on \( \mathcal{A} \). The definition is the same as in the standard setting, except that the terms “computable” and “\( \Sigma^0_\alpha \) 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 “\( \Sigma^0_\alpha \).”

**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 \( \Sigma^0_\alpha \) on \( \mathcal{A} \) if for all isomorphisms \( F \) from \( \mathcal{A} \) onto a copy \( \mathcal{B} \), \( F(R) \) is \( \Sigma^0_\alpha \) 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 \( \Sigma^0_\alpha \) on \( \mathcal{A} \),
2. \( R \) is defined in \( \mathcal{A} \) by a computable \( \Sigma_\alpha \) formula \( \varphi(\bar{c}, \bar{x}) \), with a countable tuple of parameters \( \bar{c} \).

For simplicity, we suppose that \( \mathcal{A} \) has universe equal to \( \omega_1 \), 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 \( \omega_1 \), and we let \( (\mathcal{B}, R') \equiv_F (\mathcal{A}, R) \). The forcing conditions are countable partial permutations of \( \omega_1 \). 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.

- \( b \in \mathcal{B} \),
- \( b \notin \mathcal{B} \),
- \( b \in \Delta^0_\beta(\mathcal{B}) \),
- \( b \notin \Delta^0_\beta(\mathcal{B}) \),
- \( b \in S_{(\Sigma, \beta, \gamma)}(\mathcal{B}) \),
We use a propositional language in which the propositional variables are the atomic sentences involving symbols from $L, R,$ and constants from $\omega_1$. We will identify propositional variables with their codes. The set of codes for propositional variables is a computable set. We write $\text{neg}(\varphi)$ 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 $b \in B$: 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 $b \notin B$, 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 $\Delta^0_\alpha(B)$ is just (the atomic diagram of) $B$. So, we have formulas saying that $b \in \Delta^0_\alpha(B)$ and $b \notin \Delta^0_\alpha(B)$.

- Suppose $\beta$ is a successor ordinal—$\beta = \delta + 1$. Then $\Delta^0_\beta(B)$ is the jump of $\Delta^0_\delta(B)$. To say that $b \in \Delta^0_\beta(B)$, if $b$ is a pair $(\gamma, c)$, 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 $x \in \Delta^0_\beta(B)$ if $\rho(x) = 1$ and $x \notin \Delta^0_\beta(B)$ if $\rho(x) = 0$. If $b$ is not a pair $(\gamma, c)$, we write $\bot$. To say that $b \notin \Delta^0_\beta(B)$, we apply $\text{neg}$ to the formula above.

- Suppose $\beta$ is a limit ordinal. To say that $c \in \Delta^0_\beta(B)$, if $c$ is a pair $(\delta, d)$, where $\delta < \beta$, we take the formula saying that $d \in \Delta^0_\beta(B)$, and if $c$ is not such a pair, then we write $\bot$. To say that $c \notin \Delta^0_\beta(B)$, if $c$ is a pair $(\delta, d)$, where $\delta < \beta$, then we take the formula saying that $d \notin \Delta^0_\beta(B)$, and if $c$ is not such a pair, then we write $\top$.

- To say that $b \in S_{(\Pi, \beta, \gamma)}(B)$, we take the disjunction over $\rho \in 2^{\omega_1}$ such that the triple $(\rho, \gamma, b) \in U$ of formulas saying that $c \in \Delta^0_\beta(B)$ if $\rho(c) = 1$ and $c \notin \Delta^0_\beta(B)$ if $\rho(c) = 0$.

- To say that $b \in S_{(\Pi, \beta, \gamma)}(B)$, we apply $\text{neg}$ to the formula saying that $b \in S_{(\Sigma, \alpha+1, \gamma)}$.

- To say that $R' = S_{(\Sigma, \beta, \gamma)}(B)$, we take the conjunction over all $b$ of the formulas $\bigwedge_{b}(b \in S_{(\Sigma, \beta, \gamma)}(B) \iff R'b)$.

We let $T$ include the computable $\Sigma_\beta$ and $\Pi_\beta$ formulas, for countable ordinals $\beta \leq \alpha$, plus the $\Pi_{\alpha+1}$ formulas $\chi_\gamma$ saying that $R'$ is equal to the set with index $(\Sigma, \alpha, \gamma)$ relative to $B$, and the $\Sigma_{\alpha+1}$ formulas $\text{neg}(\chi_\gamma)$. We define forcing for the formulas in $T$. 

12
Definition 17 (Definition of forcing). Let $p$ be a forcing condition. We define the relation $p \forces \varphi$, for $\varphi$ in our propositional language.

- Suppose $\varphi$ is computable $\Sigma_0$ and $\Pi_0$. Then $p \forces \varphi$ if the constants in the propositional variables that occur in $\varphi$ are all in $\text{dom}(p)$ and $p$ interprets these constants so as to make $\varphi$ true in the structure $(A, R)$.

- Suppose $\varphi$ is computable $\Sigma_\beta$, for $\beta \geq 1$, say $\varphi$ is the c.e. disjunction of formulas $\psi_i$, where $\psi_i$ is a countable conjunction of formulas $\psi_{i,j}$ and for each $j$, $\psi_{i,j}$ is $\Sigma_\gamma$ or $\Pi_\gamma$ for some $\gamma < \beta$. Then $p \forces \varphi$ if there is some $i$ such that for all $j$, $p \forces \psi_{i,j}$.

- Suppose $\varphi$ is computable $\Pi_\beta$, for $\beta \geq 1$, say $\varphi$ is the c.e. conjunction of formulas $\psi_i$, where $\psi_i$ is a countable disjunction of formulas $\psi_{i,j}$ and for each $j$, $\psi_{i,j}$ is computable $\Pi_\gamma$ or computable $\Sigma_\gamma$ for some $\gamma < \beta$. Then $p \forces \varphi$ if for all $i$ and all $q \geq p$, there exist $r \supseteq q$ and $j$ such that $r \forces \psi_{i,j}$.

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

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

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

Lemma 4.4 (Density). For all $p$ and $\varphi$, there exists $q \supseteq p$ such that $q$ “decides” $\varphi$; i.e., $q \forces \varphi$ or $q \forces \neg \varphi$.

From the definition above, forcing of computable $\Pi_\beta$ formulas does not appear to be $\Pi^0_\beta$. The lemma below gives an alternative condition, which is $\Pi^0_\beta$.

Lemma 4.5. Let $\varphi$ be a computable $\Pi_\beta$ formula, the c.e. intersection of formulas $\psi_i$, where $\psi_i$ is a countable disjunction of formulas $\psi_{i,j}$, for $j \in \omega$, each computable $\Pi_\gamma$ or computable $\Sigma_\gamma$ for some $\gamma < \beta$. For a forcing condition $p$, $p \forces \varphi$ if and only if for all $i$ and all $q \supseteq p$, it is not the case that for all $j \in \omega$, $q \forces \psi_{i,j}$.

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 $r \forces \psi_{i,j}$,

2. for all $q \supseteq p$, it is not the case that $(\forall j) \; q \forces \neg \psi_{i,j}$.

First suppose (1). If (2) fails, we would have $q \supseteq p$ such that $(\forall j) \; q \forces \neg \psi_{i,j})$. By the Extension and Consistency lemmas, we cannot have $r \supseteq q$ and $j \in \omega$ such that $r \forces \psi_{i,j}$. 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 $r \forces \psi_{i,j}$. 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 $j$, $r \forces \neg \psi_{i,j}$, 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 \(\delta\) such that \(p_\delta\) decides \(\varphi\),
3. for all \(a \in \omega_1\), there is some \(\delta\) such that \(a \in \text{ran}(p_\delta)\).

It follows from the lemmas that we can form a complete forcing sequence. For limit \(\delta\), we let \(p_\delta = \bigcup_{\varphi \in \text{dom} \varphi} p_\varphi\). Let \(F = \bigcup_\delta p_\delta\) for \(\delta < \omega_1\). From this, we obtain \(B\) and \(R'\) such that \((B,R') \equiv_F (A,R)\), as planned. Now, \(B\) and \((B,R')\) 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 \(\varphi \in T\), \((B,R') \models \varphi\) iff there is some \(\delta\) such that \(p_\delta \models \varphi\).

Proof.

1. Suppose \(\varphi\) is computable \(\Sigma_0\) and computable \(\Pi_0\). Take \(\delta\) such that \(\text{dom}(p_\delta)\) includes all of the constants that appear in the propositional variables in \(\varphi\). If \(p_\delta\) makes \(\varphi\) true in \((A,R)\), then \(p_\delta \models \varphi\) and \((B,R') \models \varphi\). If \(p_\delta\) makes \(\varphi\) false in \((A,R)\), then \(p_\delta \models \neg \varphi\) and \((B,R') \models \neg \varphi\).

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

3. Suppose \(\varphi\) is computable \(\Pi_\beta\), the c.e. conjunction of formulas \(\psi_i\), where \(\psi_i\) is a countable disjunction of formulas \(\psi_{i,j}\), each computable \(\Sigma_\gamma\) or computable \(\Pi_\gamma\), for some \(\gamma < \beta\). First, suppose \((B,R') \models \varphi\). This means that for all \(i\), \(\psi_i\) is true. For some \(\delta\), \(p_\delta\) forces either \(\varphi\) or \(\neg \varphi\). We show that \(p_\delta\) cannot force \(\neg \varphi\). If \(p_\delta \models \neg \varphi\), then there is some \(i\) such that \(p_\delta \models \neg \psi_{i,j}\). By the induction hypothesis, for all \(j\), \(p_{\delta_{i,j}} \models \neg \psi_{i,j}\) is true, so the conjunction is true, and this is \(\neg \varphi\), contradicting the fact that \(\psi_i\) is true. Now, suppose that for some \(\delta\), \(p_\delta \models \varphi\). For each \(i\) and each \(q \supseteq p_\delta\), there is some \(r \supseteq q\) such that for some \(j\), \(r \models \psi_{i,j}\). If \(\varphi\) is false, then for some \(i\), \(\psi_i\) 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 \(p_{\delta_j}\) forcing \(\neg \psi_{i,j}\). Let \(q \supseteq p_{\delta_j}\). Since \(q \supseteq p_{\delta_j}\), there is some \(r \supseteq q\) such that for some \(j\), \(r \models \psi_{i,j}\). Then \(r\) forces both \(\psi_{i,j}\) and \(\neg \psi_{i,j}\), a contradiction. Therefore, \(\varphi\) 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 $A$.

**Lemma 4.7** (Definability of forcing). For any $\varphi \in T'$, and for any tuples $\mathbf{b}$ and $\mathbf{x}$ of the same countable ordinal arity, there is a predicate formula $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x})$ such that $A = \text{Force}_{\mathbf{b}, \varphi}(\mathbf{a})$ iff the correspondence taking $b_i$ to $a_i$ is a forcing condition $p$ such that $p \models \varphi$. Moreover, if $\varphi$ is computable $\Sigma_\beta$, or computable $\Pi_\beta$, for $1 \leq \beta$, then $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x})$ is also computable $\Sigma_\beta$, or computable $\Pi_\beta$.

**Proof.** We suppose that the elements of $\mathbf{b}$ are distinct, and the variables in $\mathbf{x}$ are distinct. We have a simple formula $\text{force}_{\mathbf{b}}(\mathbf{x})$ saying that the correspondence taking $b_i$ to $x_i$ is a forcing condition—take the conjunction of formulas $x_i /\slash.left = x_j$, where $i /\slash.left = j$. Now, we give the formulas $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x})$ by induction on $\varphi$.

1. Suppose $\varphi$ is computable $\Sigma_0$ and computable $\Pi_0$. While $\varphi$ is really propositional, it is convenient to think of it also as a quantifier-free predicate sentence involving predicate symbols from the language of $A$ and constants from the universe of $\mathcal{B}$. Suppose that the constants that appear in $\varphi$ are all in $\mathbf{b}$, and let $\varphi'$ be the result of replacing each occurrence of $b_i$ in $\varphi$ by $x_i$. Then $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x}) = (\text{force}_{\mathbf{b}}(\mathbf{x}) \& \varphi')$. If the constants that appear in $\varphi$ are not all in $\mathbf{b}$, then $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x}) = \bot$. In either case, the formula $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x})$ is computable $\Sigma_0$ and computable $\Pi_0$.

2. Suppose $\varphi$ is computable $\Sigma_\beta$, the c.e. disjunction of formulas $\psi_i$, where $\psi_i$ is a countable conjunction of formulas $\psi_{i,j}$, each of which is computable $\Pi_i$, or computable $\Sigma_\gamma$, for some $\gamma < \beta$. Let $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x}) = \bigwedge_i \bigwedge_j \text{Force}_{\mathbf{b}, \psi_{i,j}}(\mathbf{x})$. Each $\text{Force}_{\mathbf{b}, \psi_{i,j}}(\mathbf{x})$ is computable $\Pi_\gamma$ or computable $\Sigma_\gamma$ for some $\gamma < \beta$. Then $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x})$ is computable $\Sigma_\alpha$.

3. Suppose that $\varphi$ is computable $\Pi_\beta$, the c.e. conjunction of formulas $\psi_i$, where $\psi_i$ is a countable disjunction of formulas $\psi_{i,j}$, each of which is computable $\Sigma_\gamma$, or computable $\Pi_\gamma$, for some $\gamma < \beta$. We think of the alternative definition of forcing for such formulas. We let $\text{Force}_{\mathbf{b}, \varphi}(\mathbf{x})$ be a computable $\Pi_\beta$ formula saying that for all $i$ and for all $\mathbf{d}$ and $\mathbf{\bar{u}}$, if the correspondence taking $b_i, \mathbf{d}$ to $\mathbf{x}, \mathbf{\bar{u}}$ is a forcing condition $q$, then it is not the case that for all $j \in \omega$, $q \models \neg \psi_{i,j}$. We write

$$\bigwedge_i \bigwedge_{\mathbf{d}} (\forall \mathbf{\bar{u}}) \left( \text{force}_{\mathbf{b}, \varphi}(\mathbf{x}, \mathbf{\bar{u}}) \rightarrow \bigwedge_j \neg \text{Force}_{\mathbf{b}, \mathbf{d}, \neg \psi_{i,j}}(\mathbf{x}, \mathbf{\bar{u}}) \right)$$

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

\[ \square \]
We are ready to complete the proof. By assumption and Lemma 4.5, some
$p$ forces $S_{(\Sigma,\alpha)}(B) = R'$, for some $\gamma$. Say that $p$ maps $\overline{d}$
to $\overline{\tau}$. We can see that
$a \in R$ iff there is some $q \supseteq p$ such that $q(b) = a$ and $q \vdash b \in S_{(\Sigma,\alpha)}(B)$. For
any $b$, the formula saying that $b \in S_{(\Sigma,\alpha)}(B)$ is computable $\Sigma_\alpha$. We can write
a computable $\Sigma_\alpha$ predicate formula $\varphi(\overline{\tau}, x)$ saying that there exists $q \supseteq p$ such
that $q(b) = x$ and $q \vdash b \in S_{(\Sigma,\alpha)}(B)$. We take the c.e. disjunction over $b, b_1$ of
the formulas $(\exists \overline{\tau})\text{Force}_{\overline{d}, b, b_1, b \in S_{(\Sigma,\alpha)}(B)}(\overline{\tau}, x, \overline{\pi})$. This formula defines $R$, and
it is computable $\Sigma_\alpha$.

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
$S_{(\Sigma,\alpha)}(B)$ is the c.e. union of the sets $S_{(\Pi, \beta, \delta)}(B)$ such that $(\Pi, \beta, \delta) \in W_\gamma$ and
$\beta < \alpha$. We write $S_{(\Sigma,\alpha)}(B)$, or $S_{(\Pi, \alpha, \gamma)}(B)$, for the set with index $(\Sigma, \alpha, \gamma)$,
or $(\Pi, \alpha, \gamma)$, relative to $B$. For $\alpha = 1$, $S_{(\Sigma, 1, \gamma)}(B) = W_\gamma^B$, and $S_{(\Pi, 1, \gamma)}(B)$
is the complementary set. For our forcing language, we need formulas with the
meanings below.

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

Again, the propositional variables are the atomic sentences involving symbols
from $L, R$, and constants from $\omega_1$.

- The formulas saying that $b \in B$ and $b \notin B$ are the same as before. To say
  that $b \in B$, we write $b$ if $b$ is a propositional variable, $\neg c$ if $b$ is the negation
  of a propositional variable $c$, and $\bot$ otherwise. To say that $b \notin B$, 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 $b \in S_{(\Sigma, 1, \gamma)}(B)$, we take the disjunction, over $\rho \in 2^{\omega}$
such that $(\rho, \gamma, b)$ is in the relation $U$, of the conjunction of formulas saying
  $x \in B$, for $\rho(x) = 1$, and formulas saying $x \notin B$, for $\rho(x) = 0$. To say that
  $b \in S_{(\Pi, 1, \gamma)}(B)$, or $b \notin S_{(\Sigma, 1, \gamma)}(B)$, we apply neg to the formula saying
  $b \in S_{(\Sigma, 1, \gamma)}(B)$.
• For $\delta > 1$, to say that $b \in S_{(\Sigma,\delta,\gamma)}(B)$, we take the disjunction, over $(\Pi,\delta',\gamma') \in W_\gamma$ with $1 \leq \delta' < \delta$, of formulas saying that $b \in S_{(\Pi,\delta',\gamma')}(B)$. To say that $b \in S_{(\Pi,\delta,\gamma)}(B)$, we apply $\text{neg}$ to the formula saying that $b \in S_{(\Sigma,\delta,\gamma)}(B)$.

• To say that $R' = S_{(\Sigma,\alpha,\gamma)}(B)$, we take the formula saying $\forall b (b \in S_{(\Sigma,\alpha,\gamma)}(B) \iff R'b)$.

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

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

• Suppose $\varphi$ is computable $\Sigma_0$ and $\Pi_0$. We say $p$ forces $\varphi$, or $p \vdash \varphi$, if the constants in the propositional variables that occur in $\varphi$ are all in $\text{dom}(p)$ and $p$ interprets these constants so as to make $\varphi$ true in $(A, R)$.

• Suppose $\varphi$ is computable $\Sigma_\beta$, for $\beta \geq 1$, a c.e. disjunction of formulas $\psi_i$, where each $\psi_i$ is computable $\Pi_\gamma$ for some $\gamma < \beta$. Then $p \vdash \varphi$ if $p \vdash \psi_i$, for some $i$.

• Suppose $\varphi$ is computable $\Pi_\beta$, for $\beta \geq 1$, a c.e. conjunction of formulas $\psi_i$, where each $\psi_i$ is computable $\Sigma_\gamma$ for some $\gamma < \beta$. Then $p \vdash \varphi$ if for all $i$ and all $q \geq p$, there is some $r \geq q$ 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 $q \geq p$, then $q \vdash \varphi$.

**Lemma 4.9** (Consistency). It is not the case that $p \vdash \varphi$ and $p \vdash \text{neg}(\varphi)$.

**Lemma 4.10** (Density). For all $p$ and $\varphi$, there exists $q \geq p$ such that $q \vdash \varphi$ or $q \vdash \text{neg}(\varphi)$.

As for the first definition, we could show that if $\varphi$ is computable $\Pi_\beta$, a c.e. conjunction of formulas $\psi_i$, each computable $\Sigma_\gamma$ for some $\gamma < \beta$, then $p \vdash \varphi$ iff for all $i$ and all $q \geq p$, $q$ does not force $\text{neg}(\psi_i)$. This is not necessary, since for the second definition, the relation $p \vdash \varphi$ is easily seen to be $\Pi_0^0$.

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

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

17
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 $\text{Force}_{\overline{b},\varphi}(\overline{x})$ such that $\mathcal{A} \models \text{Force}_{\overline{b},\varphi}(\overline{x})$ iff the correspondence taking $b_i$ to $a_i$ is a forcing condition $p$ such that $p \vdash \varphi$. Moreover, if $\varphi$ is computable $\Sigma_\beta$, or $\Pi_\beta$, for $1 \leq \beta \leq \alpha$, then $\text{Force}_{\overline{b},\varphi}(\overline{x})$ is also computable $\Sigma_\beta$, or computable $\Pi_\beta$.

**Proof.** First, suppose $\varphi$ is computable $\Sigma_0$ and computable $\Pi_0$. If $b$ includes all of the constants that appear in the propositional variables of $\varphi$, then we let $\text{Force}_{\overline{b},\varphi}(\overline{x})$ be $\text{force}_{\overline{b}}(\overline{x}) \& \varphi'$, where $\varphi'$ is the result of replacing the occurrences of the constant $b_i$ in $\varphi$ by the corresponding variable $x_i$. If $b$ does not include all of the constants of $\varphi$, then $\text{Force}_{\overline{b},\varphi}(\overline{x})$ is $1$. Suppose $\varphi$ is computable $\Sigma_\beta$, a c.e. disjunction of formulas $\psi_i$, each of which is computable $\Pi_\gamma$ for some $\gamma < \beta$. We let $\text{Force}_{\overline{b},\varphi}(\overline{x})$ be
\[
\bigvee_i \left( \text{force}_{\overline{b}}(\overline{x}) \& \text{Force}_{\overline{b},\psi_i}(\overline{x}) \right)
\]
This is a computable $\Sigma_\beta$ predicate formula. Finally, suppose $\varphi$ is computable $\Pi_\beta$, a c.e. conjunction of formulas $\psi_i$, each of which is computable $\Sigma_\gamma$, for some $\gamma < \beta$. We let $\text{Force}_{\overline{b},\varphi}(\overline{x})$ be the conjunction of $\text{force}_{\overline{b}}(\overline{x})$ and the following formulas, one for each $i$ and $\overline{v}$:
\[
(\forall \overline{v}) \left[ \text{force}_{\overline{b},\varphi}(\overline{x},\overline{v}) \rightarrow \bigwedge_i \left( (\exists \overline{v}) \left( \text{force}_{\overline{b},\varphi}(\overline{x},\overline{v}) \& \text{Force}_{\overline{b},\psi_i}(\overline{x},\overline{v}) \right) \right) \right]
\]
This is logically equivalent to a computable $\Pi_\beta$ predicate formula.

We are ready to complete the proof of the second version of the theorem. Suppose $p$ forces $S(\Sigma_\alpha,\gamma)(\overline{b}) = \overline{b}'$, where $p$ maps $\overline{d}$ to $\overline{c}$. We can see that $\alpha \in R$ iff there is some $q \supseteq p$ such that $q(b) = a$ and $q \vdash b \in S(\Sigma_\alpha,\gamma)(\overline{b})$. We have a computable $\Sigma_\alpha$ predicate formula $\varphi(\overline{x},\overline{v})$ saying that there exists $q \supseteq p$ such that $q(b) = x$ and $q \vdash b \in W_\alpha^\overline{b}$. We take the c.e. disjunction over $\overline{b},\overline{b}'$ of the formulas $(\exists \overline{v}) \text{Force}_{\overline{b},\varphi}(\overline{x},\overline{v})$. This formula defines $R$.

\[\square\]

5 \textbf{Which definition is better?}

The two definitions of the arithmetical hierarchy in the setting of $\omega_1$ are not equivalent. Each definition yields a result saying that a relation is relatively intrinsically $\Sigma_\alpha^0$ on $\mathcal{A}$ iff it is defined by a computable $\Sigma_\alpha$ 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 $\Delta_\alpha^0$ set based on finitely much $\Delta_\alpha^0$ information. We are using the full
power of the oracle. In our first definition, an element enters a $\Sigma^0_\omega$ set based on countably many pieces of $\Delta^0_\omega$ information. We may use the full power of the $\Delta^0_\omega$ oracle. In our second definition, an element enters a $\Sigma^0_\omega$ set by entering one of a c.e. family of sets, each of which is $\Pi^0_n$ for some $n < \omega$. We never use the full power of $\Delta^0_\omega$. 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


