PMATH 432: First Order Logic and Computability

Typeset by Kevin Matthews

Winter 2014
2014 05 03

1 Introduction

2 Syntax of First-Order Languages

2.1 Alphabets

Definition 1. An *alphabet* is a non-empty set of symbols.

Example 2. Examples of alphabets include $\{a, b, c, \ldots, z\}$, $\{c_0, c_1, c_2, \ldots\}$, and $\{\ddot\smile, \star, \bigstar\}$.

Definition 3. Let $\mathcal{A}$ be an alphabet. A *word* (or *string*) over $\mathcal{A}$ is a finite sequence of symbols from $\mathcal{A}$. The set of all words over $\mathcal{A}$ is denoted $\mathcal{A}^*$. The *length* of a word over $\mathcal{A}$ is the total number of symbols, counting repetition, which appear in the word. In particular, we denote the empty word (of length zero) by $\square$.

2.2 The Alphabet of a First-Order Language

Definition 4. Let $\mathbb{A}$ consist of exactly the following symbols:

(a) $v_0, v_1, v_2, \ldots$ (variables);

(b) $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ (connectives: not, and, or, implies, if and only if);

(c) $\forall, \exists$ (quantifiers: for all, there exists);

(d) $\equiv$ (equality);

(e) $), ($ (right parenthesis, left parenthesis).

A *first order language* is a set S which is disjoint from $\mathbb{A}$ and consists of:

(1) for every $n \geq 1$ a (possibly empty) set of n-ary relations symbols;

(2) for every $n \geq 1$ a set of n-ary function symbols;

(3) a set of constant symbols.

We let $\mathbb{A}_S := \mathbb{A} \cup S$. (Technically it is usually L^S, which we have yet to define, which is referred to as the first order language, as opposed to S.)

REMARK 5. As of yet, these symbols have no meaning — they are nothing more than symbols. Later we will attach meaning to them.

In general, even though the variables in a first-order language are always $v_0, v_1, v_2, \ldots$, we will often use x, y, z, v, w as variables, despite the fact that it is formally incorrect. Moreover, as convention, we will always use P, Q, R for relation symbols, use f, g, h for function symbols, and use $c, d, c_0, c_1, c_2, \ldots$ for constant symbols.

So far, any sequence of symbols from a first-order alphabet is called a word, even though many of them have no intuitive meaning (for instance, $(\equiv \neg v_0 v_{54}()$. We clearly cannot attach meaning to all words. As such, we will define sentences, formulas, and terms for a first-order language.

2.3 Terms and Formulas in First-Order Languages

DEFINITION 6. Let S be a first-order language. The following words in ${\mathbb{A}_S}^*$ are called S-*terms*:

(T1) every variable in $\mathbb{A}$ is an S-term;

(T2) every constant symbol in S is an S-term;

(T3) if $n \geq 1$, the strings $t_1, \ldots, t_n$ are S-terms, and $f \in S$ is an n-ary function symbol, then $ft_1 \cdots t_n$ is also an S-term.

We let T^S denote the set of all S-terms.

DEFINITION 7. Let S be a first-order language. The following words in ${\mathbb{A}_S}^*$ are called S-formulas:

(F1) if t_1 and t_2 are S-terms then $t_1 \equiv t_2$ is an S-formula;

(F2) if $n \geq 1$, $t_1, \ldots, t_n$ are S-terms, and $R \in S$ is a n-ary relation symbol, then $Rt_1 \cdots t_n$ is an S-formula;

(F3) if ϕ is an S-formula, then $\neg\phi$ is an S-formula;

(F4) if ϕ and ψ are S-formulas, then $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$ are also S-formulas;

(F5) if ϕ is an S-formula and x is a variable, then $\forall x\phi$ and $\exists x\phi$ are also S-formulas.

In particular, S-formulas of the form F1 and F2 are called *atomic formulas*. We let L^S denote the set of all S-formulas.

Example 8. An example of an S-formula over the first-order language with symbol set

$$\{R, f, P, c_1, c_2\},$$

where R is a 2-ary relation symbol, f is a 1-ary function symbol, P is a 1-ary relation symbol, and c_1 and c_2 are constant symbols, is

$$\exists v_{12}(\exists v_7 Rfc_1c_2 \to \forall v_8 Pv_8).$$

Remark 9. We often will use abbreviations/modified notations for formulas. For instance, if $<$ is a 2-ary relations symbol we may write $a < b$ in place of $< ab$. We will also often drop parentheses. For example, $\phi \wedge \psi \wedge \chi$ means $((\phi \wedge \psi) \wedge \chi)$. Note that we have not assigned any meaning to the symbol $\wedge$, and it is not the case that $((\phi\wedge\psi)\wedge\chi) = (\phi\wedge(\psi\wedge\chi))$, though we will see that their interpretations will be 'equivalent', in some appropriate sense.

2.4 Induction in the Calculus of Terms and in the Calculus of Formulas

Definition 10. Let S be a first-order language. The function var_S on all S-terms, which gives the set of variables occurring in the term, is defined by:

$\mathrm{var}_S(x) := \{x\}$ (where x is a variable);

$\mathrm{var}_S(c) := \varnothing$ (where c is a constant symbol);

$\mathrm{var}_S(ft_1\cdots t_n) := \mathrm{var}_S(t_1) \cup \cdots \cup \mathrm{var}_S(t_n)$ (where $n \geq 1$, $f \in S$ is an n-ary function symbol, and $t_1, \ldots, t_n$ are S-terms).

Definition 11. Let S be a first-order language. The function SF_S on all S-formulas, which gives the set of subformulas of the formula, is defined by:

$\mathrm{SF}_S(t_1 \equiv t_2) := \{t_1 \equiv t_2\}$ (where t_1 and t_2 are S-terms);

$\mathrm{SF}_S(Rt_1\cdots t_n) := \{Rt_1\cdots t_n\}$ (where $n \geq 1$, R is an n-ary relation symbol, and $t_1, \ldots, t_n$ are S-terms);

$\mathrm{SF}_S(\neg\phi) := \{\neg\phi\} \cup \mathrm{SF}_S(\phi)$ (where ϕ is an S-formula);

$\mathrm{SF}_S((\phi\bigstar\psi)) := \{(\phi\bigstar\psi)\} \cup \mathrm{SF}_S(\phi) \cup \mathrm{SF}_S(\psi)$ for all $\bigstar \in \{\vee, \wedge, \to, \leftrightarrow\}$ (where ϕ and ψ are S-formulas);

$SF_S(Qx\phi) := \{Qx\phi\} \cup SF_S(\phi)$ for all $Q \in \{\forall, \exists\}$ (where x is a variable and ϕ is an S-formula).

2.5 Free Variables and Sentences

DEFINITION 12. Let S be a first-order language. The function free_S on all S-formulas, which gives the set of free variables in the formula, is defined by:

$\mathrm{free}_S(t_1 \equiv t_2) := \mathrm{var}_S(t_2) \cup \mathrm{var}_S(t_2)$ (where t_1 and t_2 are S-terms);

$\mathrm{free}_S(Rt_1 \cdots t_n) := \mathrm{var}_S(t_1) \cup \cdots \cup \mathrm{var}_S(t_n)$ (where $n \geq 1$, $R \in S$ is an n-ary relation symbol, and $t_1, \ldots, t_n$ are S-terms);

$\mathrm{free}_S(\neg\phi) := \mathrm{free}_S(\phi)$ (where ϕ is an S-formula);

$\mathrm{free}_S((\phi \bigstar \psi)) := \mathrm{free}_S(\phi) \cup \mathrm{free}_S(\psi)$ for all $\bigstar \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ (where ϕ and ψ are S-formulas);

$\mathrm{free}_S(Qx\phi) := \mathrm{free}_S(\phi) \setminus \{x\}$ for all $Q \in \{\forall, \exists\}$ (where x is a variable and ϕ is an S-formula).

DEFINITION 13. Let S be a first-order language, and let ϕ be an S-formula. We say ϕ is an S-*sentence* if $\mathrm{free}(\phi) = \varnothing$.

DEFINITION 14. Let S be a first-order language, and let $n \in \mathbb{N}$. We define

$$L_n^S := \{\phi \in L^S ; \mathrm{free}_S(\phi) \subseteq \{v_0, \ldots, v_{n-1}\}\}.$$

In particular, L_0^S is the set of S-sentences.

3 Semantics of First-Order Languages

3.1 Structure and Interpretations

DEFINITION 15. Let S be a first-order language. An S-*structure* is a pair $(A, \mathfrak{a})$, where A is a non-empty set and $\mathfrak{a}$ is a map on S such that:

(1) for every $n \geq 1$ and every n-ary relation symbol $R \in S$, $\mathfrak{a}(R)$ is an n-ary relation on A;

(2) for every $n \geq 1$ and every n-ary function symbol $f \in S$, $\mathfrak{a}(f)$ is an n-ary function on A;

(3) for every constant symbol $c \in S$, $\mathfrak{a}(c)$ is an element of A.

We call A the *domain* or *universe* of $(A, \mathfrak{a})$.

REMARK 16. We will usually write $R^{\mathfrak{A}}$ in place of $\mathfrak{a}(R) \subseteq A^n$, $f^{\mathfrak{A}}$ in place of $\mathfrak{a}(f) : A^n \to A$, and $c^{\mathfrak{A}}$ in place of $\mathfrak{a}(c) \in A$, where $\mathfrak{A} = (A, \mathfrak{a})$.

EXAMPLE 17. Let $S = \{R, c, d\}$, where R is a 2-ary relation symbol and c and d are constant symbols. Consider the S-structure $\mathfrak{A} = (\mathbb{N}, \mathfrak{a})$, where $R^{\mathfrak{A}} = \{(a, b); a \le b\}$, $c^{\mathfrak{A}} = 0$, and $d^{\mathfrak{A}} = 1$. We think that Rcd and $\forall x \exists y Rxy$ should be "true in $\mathfrak{A}$", while $\exists y Rxy$ should be meaningless in $\mathfrak{A}$. We need to define "true in $\mathfrak{A}$".

DEFINITION 18. Let S be a first-order language, and let $(A, \mathfrak{a})$ be an S-structure. An *assignment* in $(A, \mathfrak{a})$ is a map $\beta : \{v_n; n \in \mathbb{N}\} \to A$.

DEFINITION 19. Let S be a first-order language, let $\mathfrak{A}$ be an S-structure, and let β be an assignment in $\mathfrak{A}$. The pair $(\mathfrak{A}, \beta)$ is called an *S-interpretation.*

DEFINITION 20. Let S be a first-order language, let $(A, \mathfrak{a})$ be an S-structure, and let β be an assignment in $(A, \mathfrak{a})$. For $a \in A$ and a variable x, we let $\beta\frac{a}{x}$ be the assignment in $(A, \mathfrak{a})$ given by

$$\beta\frac{a}{x}(v_i) := \begin{cases} \beta(v_i) & \text{if } x \neq v_i \\ a & \text{if } x = v_i \end{cases},$$

and we let $((A, \mathfrak{a}), \beta)\frac{a}{x}$ be the S-interpretation $((A, \mathfrak{a}), \beta\frac{a}{x})$.

3.2 Standardization of Connectives

3.3 The Satisfaction Relation

DEFINITION 21. Let S be a first-order language, and let $\mathfrak{I} = (\mathfrak{A}, \beta)$ be an S-interpretation, where $\mathfrak{A} = (A, \mathfrak{a})$. To every S-term t, we associate an element $\mathfrak{I}(t) \in A$ as follows:

(a) $\mathfrak{I}(x) := \beta(x)$ (where x is a variable);

(b) $\mathfrak{I}(c) := \mathfrak{a}(c)$ (where c is a constant symbol);

(c) $\mathfrak{I}(ft_1 \cdots ft_n) := \mathfrak{a}(f)(\mathfrak{I}(t_1), \ldots, \mathfrak{I}(t_n))$ (where $n \ge 1$, $f \in S$ is an n-ary function symbol, and $t_1, \ldots, t_n$ are S-terms).

DEFINITION 22. Let S be a first-order language, and let $\mathfrak{I} = ((A, \mathfrak{a}), \beta)$ be an S-interpretation. For an S-formula ϕ, we say $\mathfrak{I}$ *is a model of* ϕ (or $\mathfrak{I}$ *satisfies* ϕ, or ϕ *holds in* $\mathfrak{I}$), and write $\mathfrak{I} \models \phi$, in the following situations:

$\mathfrak{I} \models t_1 \equiv t_2$ if $\mathfrak{I}(t_1) = \mathfrak{I}(t_2)$ (where t_1 and t_2 are S-terms);

$\mathfrak{I} \models Rt_1 \cdots t_n$ if $(\mathfrak{I}(t_1), \ldots, \mathfrak{I}(t_n)) \in \mathfrak{a}(R)$ (where $n \geq 1$, $R \in S$ is an n-ary relation symbol, and $t_1, \ldots, t_n$ are S-terms);

$\mathfrak{I} \models \neg\phi$ if not $\mathfrak{I} \models \phi$ (where ϕ is an S-formula);

$\mathfrak{I} \models (\phi \vee \psi)$ if $\mathfrak{I} \models \phi$ or $\mathfrak{I} \models \psi$ (where ϕ and ψ are S-formulas);

$\mathfrak{I} \models (\phi \wedge \psi)$ if $\mathfrak{I} \models \phi$ and $\mathfrak{I} \models \psi$ (where ϕ and ψ are S-formulas);

$\mathfrak{I} \models (\phi \rightarrow \psi)$ if if $\mathfrak{I} \models \phi$ then $\mathfrak{I} \models \psi$ (where ϕ and ψ are S-formulas);

$\mathfrak{I} \models (\phi \leftrightarrow \psi)$ if $\mathfrak{I} \models \phi$ if and only if $\mathfrak{I} \models \psi$ (where ϕ and ψ are S-formulas);

$\mathfrak{I} \models \forall x\phi$ if for all $a \in A$ we have $\mathfrak{I}\frac{a}{x} \models \phi$ (where ϕ is an S-formula and x is a variable);

$\mathfrak{I} \models \exists x\phi$ if there is an $a \in A$ such that $\mathfrak{I}\frac{a}{x} \models \phi$ (where ϕ is an S-formula and x is a variable).

Moreover, for $\Phi \subseteq L^S$ we write $\mathfrak{I} \models \Phi$ if $\mathfrak{I} \models \phi$ for all $\phi \in \Phi$.

3.4 The Consequence Relation

Definition 23. Let S be a first-order language, and let $\mathfrak{I}$ be an S-interpretation, let $\phi \in L^S$ be an S-formula, and let $\Phi \subseteq L^S$ be a set of S-formulas. We say ϕ *is a consequence of* Φ, and write $\Phi \models_S \phi$, if for all S-interpretations $\mathfrak{J}$ such that $\mathfrak{J} \models \Phi$, we have $\mathfrak{J} \models \phi$. (We will see later that this is independent of S, via the coincidence lemma.)

Example 24. For an S-formula ϕ, $\{(\phi \wedge \phi)\} \models_S \phi$. Indeed, if $\mathfrak{I} \models (\phi \wedge \phi)$, then by definition we have $\mathfrak{I} \models \phi$ and $\mathfrak{I} \models \phi$, so in particular we know $\mathfrak{I} \models \phi$. As $\mathfrak{I}$ was arbitrary, we have $\Phi \models_S \phi$.

Definition 25. Let S be a first-order language, and let ϕ be an S-formula. We say that ϕ is *valid*, and write $\models_S \phi$, if $\varnothing \models_S \phi$.

Example 26. Examples of valid formulas include $(\phi \vee \neg\phi)$ and $\exists x x \equiv x$. The second follows because the universe A is non-empty. So if we have any interpretation $\mathfrak{I}$, we can take $\mathfrak{I}\frac{a}{x}$, for some $a \in A$, to be such that $\mathfrak{I}\frac{a}{x} \models x \equiv x$.

Definition 27. Let S be a first-order language, let ϕ be an S-formula, and let Φ be a set of S-formulas. We say ϕ is *satisfiable*, and write $\mathrm{Sat}_S\, \phi$, if there exists an S-interpretation

$\mathfrak{I}$ such that $\mathfrak{I} \models \phi$. We say Φ is *satisfiable*, and write $\mathrm{Sat}_S\,\Phi$, if there exists an S-interpretation $\mathfrak{I}$ such that $\mathfrak{I} \models \Phi$. (We will see that this is indeed independent of S, via the coincidence lemma.)

REMARK **28.** We see that ϕ is satisfiable if and only if $\neg\phi$ is not valid.

LEMMA **29.** *Let S be a first-order language, let ϕ be an S-formula, and let Φ be a set of S-formulas. Then we have $\Phi \models_S \phi$ if and only if we do not have $\mathrm{Sat}_S\,\Phi \cup \{\neg\phi\}$.*

Proof. We have $\Phi \models_S \phi$ if and only if for all S-interpretations $\mathfrak{I}$ such that $\mathfrak{I} \models \Phi$, we have $\mathfrak{I} \models \phi$. But this holds if and only if there is no S-interpretation $\mathfrak{I}$ such that $\mathfrak{I} \models \Phi$ while not $\mathfrak{I} \models \phi$. But this holds if and only if there is no S-interpretation $\mathfrak{I}$ such that $\mathfrak{I} \models \Phi$ and $\mathfrak{I} \models \neg\phi$, or equivalently, $\mathfrak{I} \models \Phi \cup \{\neg\phi\}$. But this holds if and only if we do not have $\mathrm{Sat}_S\,\Phi \cup \{\neg\phi\}$. ∎

DEFINITION **30.** Let S be a first-order language, and let ϕ and ψ be S-formulas. We say that ϕ and ψ are *logically equivalent*, and write $\phi \mathrel{=\!\!\mid\!\models_S} \psi$, if $\{\phi\} \models_S \psi$ and $\{\psi\} \models_S \phi$.

REMARK **31.** We see that $\phi \mathrel{=\!\!\mid\!\models_S} \psi$ if and only if $(\phi \leftrightarrow \psi)$ is valid.

PROPOSITION **32.** *Let S be a first-order language, let x be a variable, and let ϕ and ψ be S-formulas. Then:*

(1) $(\phi \wedge \psi) \mathrel{=\!\!\mid\!\models_S} \neg(\neg\phi \vee \neg\psi)$;

(2) $(\phi \to \psi) \mathrel{=\!\!\mid\!\models_S} (\neg\phi \vee \psi)$;

(3) $(\phi \leftrightarrow \psi) \mathrel{=\!\!\mid\!\models_S} (\neg(\phi \vee \psi) \vee \neg(\neg\phi \vee \neg\psi))$;

(4) $\forall x\phi \mathrel{=\!\!\mid\!\models_S} \neg\exists x\neg\phi$.

Proof. Exercise. ∎

REMARK **33.** Thus the connectives $\wedge$, $\to$, and $\leftrightarrow$, and the quantifier $\forall$, are superfluous (in the sense that given a formula, we can find a logically equivalent formula containing none of these symbols). We will no longer consider them part of our language (that is, we change the definition of $\mathbb{A}$), but will continue to use them as shorthand.

(Though in reality we should keep them in our language and be aware that we can always find a logically equivalent formula which contains none of them, given any formula possibly containing them. Unfortunately the text we are following does not agree with us on this.)

Lemma 34. [Coincidence Lemma] *Let S_1 and S_2 be first-order languages, let $t \in T^{S_1 \cap S_2}$ be an $S_1 \cap S_2$-term, and let $\phi \in L^{S_1 \cap S_2}$ be an $S_1 \cap S_2$-formula. For any S_1-interpretation $\mathfrak{I}_1 = ((A_1, \mathfrak{a}_1), \beta_1)$ and any S_2-interpretation $\mathfrak{I}_2 = ((A_2, \mathfrak{a}_2), \beta_2)$ such that $A_1 = A_2$:*

(a) If $\mathfrak{a}_1$ and $\mathfrak{a}_2$ agree on the symbols occurring in t and if β_1 and β_2 agree on the symbols occuring in $\mathrm{var}_{S_1 \cap S_2}(t)$, then $\mathfrak{I}_1(t) = \mathfrak{I}_2(t)$.

(b) If $\mathfrak{a}_1$ and $\mathfrak{b}_2$ agree on the symbols occurring in ϕ and if β_1 and β_2 agree on $\mathrm{free}_{S_1 \cap S_2}(\phi)$, then $\mathfrak{I}_1 \models \phi$ if and only if $\mathfrak{I}_2 \models \phi$.

Proof. (a) This part is left as an exercise.

(b) We proceed by induction on ϕ.

First suppose $\phi = t_1 \equiv t_2$ for $S_1 \cap S_2$-terms t_1 and t_2. Suppose $\mathfrak{I}_1$ and $\mathfrak{I}_2$ are such interpretations. Since $\mathfrak{I}_1$ and $\mathfrak{I}_2$ agree on the symbols in ϕ, they agree on the symbols in t_1. Since $\mathfrak{I}_1$ and $\mathfrak{I}_2$ agree on $\mathrm{free}_{S_1 \cap S_2}(t_1 \equiv t_2)$, they agree on $\mathrm{var}_{S_1 \cap S_2}(t_1)$ (indeed, $\mathrm{var}_{S_1 \cap S_2}(t_1) \subseteq \mathrm{free}_{S_1 \cap S_2}(t_1 \equiv t_2)$). Hence $\mathfrak{I}_1(t_1) = \mathfrak{I}_2(t_1)$. Similarly, $\mathfrak{I}_1(t_2) = \mathfrak{I}_2(t_2)$. So $\mathfrak{I}_1 \models t_1 \equiv t_2$ if and only if $\mathfrak{I}_1(t_1) = \mathfrak{I}_1(t_2)$ if and only if $\mathfrak{I}_2(t_1) = \mathfrak{I}_2(t_2)$ if and only if $\mathfrak{I}_2 \models t_1 \equiv t_2$.

The cases $Rt_1 \cdots t_n$, $\neg\psi$, and $(\psi \vee \chi)$ are equally straightforward.

Now suppose $\phi = \exists x \psi$, where the result holds for ψ. Suppose $\mathfrak{I}_1$ and $\mathfrak{I}_2$ are such interpretations. Note that for $\mathrm{free}_{S_1 \cap S_2}(\psi) \subseteq \mathrm{free}_{S_1 \cap S_2}(\exists x \psi) \cup \{x\}$. Now for any $a \in A$, $\mathfrak{I}_1 \frac{a}{x}$ and $\mathfrak{I}_2 \frac{a}{x}$ agree on x certainly, and agree on $\mathrm{free}_{S_1 \cap S_2}(\exists x \phi)$ by assumption. Hence they would agree on $\mathrm{free}_{S_1 \cap S_2}(\psi)$. We would then be able to apply the inductive hypothesis to conclude that $\mathfrak{I}_1 \frac{a}{x} \models \psi$ if and only if $\mathfrak{I}_2 \frac{a}{x} \models \psi$. Hence $\mathfrak{I}_1 \models \exists x \psi$ if and only if there is an $a \in A$ such that $\mathfrak{I}_1 \frac{a}{x} \models \psi$ if and only if there is an $a \in A$ such that $\mathfrak{I}_2 \frac{a}{x} \models \psi$ if and only if $\mathfrak{I}_2 \models \exists x \psi$.

Thus the induction holds.

∎

Remark 35. Now that we have the coincidence lemma, we can show that many of our previous notions are actually independent of the underlying language. Namely, logical consequence, logical equivalence, satisfaction, and validity are all independent of the language. This is made precise in the next corollary (except for satisfaction, which is discussed after reducts below).

Corollary 36. *Let S_1 and S_2 be first-order languages, let $\phi, \psi \in L^{S_1 \cap S_2}$, and let $\Phi \subseteq L^{S_1 \cap S_2}$.*

(1) Then $\Phi \models_{S_1} \phi$ if and only if $\Phi \models_{S_2} \phi$.

(2) Then $\phi \mathrel{=\!\!|\!\models}_{S_1} \psi$ if and only if $\phi \mathrel{=\!\!|\!\models}_{S_2} \psi$.

Proof. Exercise. ∎

Definition 37. Let S be a first-order language, let $\mathfrak{A} = (A, \mathfrak{a})$ be an S-structure, and let $\phi \in L_n^S$. We write $\mathfrak{A} \models \phi[a_0, \ldots, a_{n-1}]$ for $a_0, \ldots, a_{n-1} \in A$ if $(\mathfrak{A}, \beta) \models \phi$ for some assignment β in $\mathfrak{A}$ such that $\beta(v_i) = a_i$ for $i \in \{0, \ldots, n-1\}$. If ϕ is an S-sentence then we will write $\mathfrak{A} \models \phi$ in place of $\mathfrak{A} \models \phi[]$. For an S-term t with $\mathrm{var}_S(t) \subseteq \{v_0, \ldots, v_{n-1}\}$ and $a_0, \ldots, a_{n-1} \in A$, we write $t^{\mathfrak{A}}[a_0, \ldots, a_{n-1}]$ in place of $(\mathfrak{A}, \beta)(t)$, where β is an assignment in $\mathfrak{A}$ such that $\beta(v_i) = a_i$ for $i \in \{0, \ldots, n-1\}$.

Remark 38. Note that these definitions are independent of β, in the sense that if we have two such assignments β_1 and β_2, then $(\mathfrak{A}, \beta_1) \models \phi$ if and only if $(\mathfrak{A}, \beta_2) \models \phi$. This follows from the coincidence lemma (details as an exercise).

Definition 39. Let S and S' be first order languages with $S \subseteq S'$. Let $(A, \mathfrak{a})$ be an S-structure and let $(A', \mathfrak{a}')$ be an S'-structure. We say that $(A, \mathfrak{a})$ is a *reduct* of $(A', \mathfrak{a}')$ and $(A', \mathfrak{a}')$ is an *expansion* of $(A, \mathfrak{a})$ if $A = A'$ and if $\mathfrak{a}$ and $\mathfrak{a}'$ agree on S.

Remark 40. By the coincidence lemma, if ϕ is an S-formula with $\phi \in L_n^S$ then for any $a_0, \ldots, a_{n-1} \in A$ we have $\mathfrak{A} \models \phi[a_0, \ldots, a_{n-1}]$ if and only if $\mathfrak{A}' \models \phi[a_0, \ldots, a_{n-1}]$ (where $\mathfrak{A}$ is a reduct of $\mathfrak{A}'$).

Moreover, we can now show that satisfaction is independent of the underlying language.

Corollary 41. *Let S_1 and S_2 be first-order languages and let $\Phi \subseteq L^{S_1 \cap S_2}$. Then* $\mathrm{Sat}_{S_1} \Phi$ *if and only if* $\mathrm{Sat}_{S_2} \Phi$.

Proof. Exercise. ∎

3.5 Two Lemmas on the Satisfaction Relation

Definition 42. Let S be a first-order language and let $(A, \mathfrak{a})$ and $(B, \mathfrak{b})$ be S-structures. We say a map $\pi : A \to B$ is an *isomorphism from* $(A, \mathfrak{a})$ *to* $(B, \mathfrak{b})$, and write $\pi : (A, \mathfrak{a}) \cong (B, \mathfrak{b})$ if:

(1) π is a bijection of A onto B;

(2) for every $n \geq 1$, every n-ary relation symbol $R \in S$, and every $a_0, \ldots, a_{n-1} \in A$, $(a_0, \ldots, a_{n-1}) \in \mathfrak{a}(R)$ if and only if $(\pi(a_0), \ldots, \pi(a_{n-1})) \in \mathfrak{b}(R)$;

(3) for every $n \geq 1$, every n-ary function symbol $f \in S$, and every $a_0, \ldots, a_{n-1} \in A$, $\pi((\mathfrak{a}(f))((a_0, \ldots, a_{n-1}))) = (\mathfrak{b}(f))((\pi(a_0), \ldots, \pi(a_{n-1}))$;

(4) for every constant symbol $c \in S$, $\pi(\mathfrak{a}(c)) = \mathfrak{b}(c)$.

We say $(A, \mathfrak{a})$ and $(B, \mathfrak{b})$ are *isomorphic* if there exists an isomorphism from $(A, \mathfrak{a})$ to $(B, \mathfrak{b})$, and in this case we write $(A, \mathfrak{a}) \cong (B, \mathfrak{b})$.

LEMMA 43. [ISOMORPHISM LEMMA] *Let S be a first-order language and let $\mathfrak{A} = (A, \mathfrak{a})$ and $\mathfrak{B} = (B, \mathfrak{b})$ be S-structures. If $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic then for all S-sentences ϕ we have $\mathfrak{A} \vDash \phi$ if and only if $\mathfrak{B} \vDash \phi$.*

Proof. As $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic, there exists an isomorphism $\pi : A \to B$. For any S-interpretation $(\mathfrak{A}, \beta)$ we have a naturally induced S-interpretation $(\mathfrak{B}, \pi \circ \beta)$.

First we will show that for every S-term t and every assignment β in $\mathfrak{A}$, $\pi((\mathfrak{A}, \beta)(t)) = (\mathfrak{A}, \pi \circ \beta)(t)$. We proceed by induction on S-terms. If x is a variable and β is an assignment in $\mathfrak{A}$, then

$$\pi((\mathfrak{A}, \beta)(x)) = \pi(\beta(x)) = (\pi \circ \beta)(x) = (\mathfrak{B}, \pi \circ \beta)(x).$$

If $c \in S$ is a constant symbol and β is an assignment in $\mathfrak{A}$, then

$$\pi((\mathfrak{A}, \beta)(c)) = \pi(\mathfrak{a}(c)) = \mathfrak{b}(c) = (\mathfrak{B}, \pi \circ \beta)(c)$$

since π is an isomorphism. If $f \in S$ is an n-ary function, $t_0, \ldots, t_{n-1}$ are S-terms for which the result holds, and β is an assignment in $\mathfrak{A}$, then

$$\begin{aligned}
\pi((\mathfrak{A}, \beta)(ft_0 \cdots t_{n-1})) &= \pi((\mathfrak{a}(f))(((\mathfrak{A}, \beta)(t_0), \ldots, (\mathfrak{A}, \beta)(t_{n-1})))) \\
&= (\mathfrak{b}(f))((\pi((\mathfrak{A}, \beta)(t_0)), \ldots, \pi((\mathfrak{A}, \beta)(t_{n-1})))) \\
&= (\mathfrak{b}(f))(((\mathfrak{A}, \pi \circ \beta)(t_0), \ldots, (\mathfrak{A}, \pi \circ \beta)(t_{n-1}))) \\
&= (\mathfrak{B}, \pi \circ \beta)(ft_0 \cdots t_{n-1}),
\end{aligned}$$

because π is an isomorphism and by inductive assumption.

Next we will show that for every S-formula ϕ, for all assignments β in $\mathfrak{A}$ we have $(\mathfrak{A}, \beta) \vDash \phi$ if and only if $(\mathfrak{A}, \pi \circ \beta) \vDash \phi$. We proceed by induction on S-formulas. If t_0 and t_1 are S-terms and β is an assignment in $\mathfrak{A}$, then

$$\begin{aligned}
(\mathfrak{A}, \beta) \vDash t_0 \equiv t_1 &\text{ if and only if } (\mathfrak{A}, \beta)(t_0) = (\mathfrak{A}, \beta)(t_1) && \text{(definition)} \\
&\text{ if and only if } \pi((\mathfrak{A}, \beta)(t_0)) = \pi((\mathfrak{A}, \beta)(t_1)) && (\pi \text{ is injective}) \\
&\text{ if and only if } (\mathfrak{A}, \pi \circ \beta)(t_0) = (\mathfrak{A}, \pi \circ \beta)(t_1) && \text{(just saw)} \\
&\text{ if and only if } (\mathfrak{A}, \pi \circ \beta) \vDash t_0 \equiv t_1.
\end{aligned}$$

The cases $Rt_0 \cdots t_{n-1}$, $\neg\psi$, and $\psi \vee \chi$ are left as similar exercises. If ψ is an S-formula for which the result holds, x is a variable, and β is an assignment in $\mathfrak{A}$, then

$$\begin{aligned}
(\mathfrak{A}, \beta) \vDash \exists x\psi &\text{ if and only if there is an } a \in A \text{ such that } (\mathfrak{A}, \beta\tfrac{a}{x}) \vDash \psi && \text{(definition)} \\
&\text{ if and only if there is an } a \in A \text{ such that } (\mathfrak{B}, \pi \circ (\beta\tfrac{a}{x})) \vDash \psi && \text{(inductive assumption)} \\
&\text{ if and only if there is an } a \in A \text{ such that } (\mathfrak{B}, (\pi \circ \beta)\tfrac{\pi(a)}{x}) \vDash \psi && \text{(definition)} \\
&\text{ if and only if there is a } b \in B \text{ such that } (\mathfrak{B}, (\pi \circ \beta)\tfrac{b}{x}) \vDash \psi && (\pi \text{ is surjective}) \\
&\text{ if and only if } (\mathfrak{B}, \pi \circ \beta) \vDash \exists x\psi.
\end{aligned}$$

In particular, if we apply the last result to S-sentences ϕ, we see that $(\mathfrak{A}, \beta) \vDash \phi$ if and only if $(\mathfrak{B}, \pi \circ \beta) \vDash \phi$. But this means, since ϕ is a sentence, that $\mathfrak{A} \vDash \phi$ if and only if $\mathfrak{B} \vDash \phi$, as required. ∎

Definition 44. Let S be a first-order language and let $(A, \mathfrak{a})$ and $(B, \mathfrak{b})$ be S-structures. We say $(A, \mathfrak{a})$ is a *substructure* of $(B, \mathfrak{b})$, and write $(A, \mathfrak{a}) \subseteq (B, \mathfrak{b})$, if:

$A \subseteq B$;

for $n \geq 1$ and n-ary relation symbols $R \in S$ we have $\mathfrak{a}(R) = \mathfrak{b}(R) \cap A^n$;

for $n \geq 1$ and n-ary function symbols $f \in S$ we have $\mathfrak{a}(f) = \mathfrak{b}(f)|_A$;

and for constant symbols $c \in S$ we have $\mathfrak{a}(c) = \mathfrak{b}(c)$.

Example 45. Let $\mathfrak{a}(+)$ be the usual addition on $\mathbb{N}$, let $\mathfrak{a}(0)$ be the natural number zero, let $\mathfrak{b}(+)$ be the usual addition on $\mathbb{Z}$, and let $\mathfrak{b}(0)$ be the integer zero. Then, for example, $(\mathbb{N}, littlea)$ is a substructure of $(\mathbb{Z}, littleb)$ (as $\{+, 0\}$-structures, with $+$ a binary function symbol and 0 a constant symbol, of course).

3.6 Simple Formalizations

3.7 Some Remarks on Formalizability

3.8 Substitution

Remark 46. It is often convenient to substitute terms for variables, in a given term or formula, as we will see later.

Definition 47. Let S be a first-order language and let t be an S-term. Let $r \in \mathbb{N}$, let $x_0, \ldots, x_r$ be pairwise distinct variables, and let $t_0, \ldots, t_r$ be S-terms. We define

$$[t]\frac{t_0 \cdots t_r}{x_0 \cdots x_r}$$

as follows:

- If t is a variable then

$$[t]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := \begin{cases} t & \text{if } t \notin \{x_0, \ldots, x_r\} \\ t_i & \text{if } t = x_i \end{cases};$$

- If t is a constant symbol then

$$[t]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := t;$$

- If $t = ft_1' \cdots t_n'$ for n-ary function symbol $f \in S$ and S-terms $t_1', \dots, t_n'$, then

$$[t]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := f[t_1']\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \cdots [t_n']\frac{t_0 \cdots t_r}{x_0 \cdots x_r}.$$

Definition 48. Let S be a first-order language and let ϕ be an S-formula. Let $r \in \mathbb{N}$, let $x_0, \dots, x_r$ be pairwise distinct variables, and let $t_0, \dots, t_r$ be S-terms. We define

$$[\phi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r}$$

as follows:

- If $\phi = t_1' \equiv t_2'$ for S-terms t_1' and t_2', then

$$[\phi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := [t_1']\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \equiv [t_2']\frac{t_0 \cdots t_r}{x_0 \cdots x_r};$$

- If $\phi = Rt_1' \cdots t_n'$ for n-ary relation symbol $R \in S$ and S-terms $t_1', \dots, t_n'$, then

$$[\phi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := R[t_1']\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \cdots [t_n']\frac{t_0 \cdots t_r}{x_0 \cdots x_r};$$

- If $\phi = \neg\psi$ for an S-formula ψ, then

$$[\phi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := \neg[\psi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r};$$

- If $\phi = \psi \vee \chi$ for S-formulas ψ and χ, then

$$[\phi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := [\psi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \vee [\chi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r};$$

- If $\phi = \exists x\psi$ for a variable x and an S-formula ψ, then let $x_{i_1}, \dots, x_{i_\ell}$ be the variables such that $x_{i_j} \in \mathrm{free}_S(\exists x\psi)$ and $x_{i_j} \neq t_{i_j}$. Let

$$u := \begin{cases} x & \text{if } x \notin \{t_{i_1}, \dots, t_{i_\ell}\}, \\ v_{\min\{j \in \mathbb{N};\, v_j \notin \mathrm{var}_S(t_{i_1}) \cup \cdots \cup \mathrm{var}_S(t_{i_\ell}) \text{ and } v_j \text{ does not occur in } \psi\}} & \text{otherwise,} \end{cases}$$

  and define

$$[\phi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} := \exists u[\psi]\frac{t_{i_1} \cdots t_{i_\ell} u}{x_{i_1} \cdots x_{i_\ell} x}.$$

Example 49. For example, if R is a ternary relation symbol, f is a unary function symbol, and c is a constant symbol, then

$$[\exists v_0 R v_0 v_1 v_6]\frac{v_1 v_0 fc}{v_0 v_1 v_6} = \exists v_2 [R v_0 v_1 v_6]\frac{v_0 fc v_2}{v_1 v_6 v_0} = \exists v_2 R v_2 v_0 fc.$$

Definition 50. Let S be a first-order language. Let $r \in \mathbb{N}$, let $x_0, \ldots, x_r$ be pairwise distinct variables, let $a_0, \ldots, a_r \in A$, and let $((A, \mathfrak{a}), \beta)$ be an S-interpretation. We define

$$\beta\frac{a_0 \cdots a_r}{x_0 \cdots x_r}(y) := \begin{cases} \beta(y) & \text{if } y \notin \{x_0, \ldots, x_r\} \\ a_i & \text{if } y = x_i, \end{cases}$$

and

$$((A, \mathfrak{a}), \beta)\frac{a_0 \cdots a_r}{x_0 \cdots x_r} := ((A, \mathfrak{a}), \beta\frac{a_0 \cdots a_r}{x_0 \cdots x_r}).$$

Lemma 51. [Substitution Lemma] *Let S be a first-order language, let $r \in \mathbb{N}$, let $t_0, \ldots, t_r$ be S-terms, and let $x_0, \ldots, x_r$ be pairwise distinct variables. If t is an S-term and ϕ is an S-formula then for all S-interpretations $\mathfrak{I} = ((A, \mathfrak{a}), \beta)$ we have*

$$\mathfrak{I}([t]\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) = \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(t)$$

and

$$\mathfrak{I} \models [\phi]\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \quad \textit{if and only if} \quad \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r} \models \phi.$$

Proof. The first part of the proof is by induction on terms t. Suppose first $t = v_i$ is a variable. If $v_i = x_j$ for some $j \in \{0, \ldots, r\}$ then

$$\mathfrak{I}([x_j]\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) = \mathfrak{I}(t_j) = \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(x_j).$$

Otherwise,

$$\mathfrak{I}([t]\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) = \mathfrak{I}(t) = \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(t).$$

Suppose that $t = c$ is a constant symbol. Then

$$\mathfrak{I}([c]\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) = \mathfrak{I}(c) = \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(c).$$

Now suppose $t = ft_1' \cdots t_\ell'$ for ℓ-ary function symbol f and terms $t_0', \cdots, t_\ell'$, and suppose the result holds for these terms. Then

$$\begin{aligned} \mathfrak{I}([ft_0' \cdots t_\ell']\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) &= \mathfrak{I}(f[t_0']\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \cdots [t_\ell']\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) \\ &= f(\mathfrak{I}([t_0']\frac{t_0 \cdots t_r}{x_0 \cdots x_r}), \ldots, \mathfrak{I}([t_\ell']\frac{t_0 \cdots t_r}{x_0 \cdots x_r})) \\ &= f(\mathfrak{I}\frac{(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(t_0'), \ldots, \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(t_\ell')) \\ &= \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(ft_0' \cdots t_\ell'). \end{aligned}$$

This completes the first part of the proof.

The second part of the proof is by induction on formulas ϕ. Suppose $\phi = t_1' \equiv t_2'$ for terms t_1' and t_2'. Then by the first part of the lemma, we have

$$\begin{aligned} \mathfrak{I} \models [t_1' \equiv t_2']\frac{t_0 \cdots t_r}{x_0 \cdots x_r} &\text{ if and only if } \mathfrak{I} \models [t_1']\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \equiv [t_2']\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \\ &\text{ if and only if } \mathfrak{I}([t_1']\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) = \mathfrak{I}([t_2']\frac{t_0 \cdots t_r}{x_0 \cdots x_r}) \\ &\text{ if and only if } \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(t_1') = \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r}(t_2') \\ &\text{ if and only if } \mathfrak{I}\frac{\mathfrak{I}(t_0) \cdots \mathfrak{I}(t_r)}{x_0 \cdots x_r} \models t_1' \equiv t_2'. \end{aligned}$$

Next suppose $\phi = Rt'_1\cdots t'_\ell$ for ℓ-ary relation symbol R and terms $t'_1,\ldots,t'_\ell$. Then we see

$$\begin{aligned}
\mathfrak{I} \models [Rt'_1\cdots t'_\ell]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} &\text{ if and only if } \mathfrak{I} \models R[t'_1]\tfrac{t_0\cdots t_r}{x_0\cdots x_r}\cdots[t'_\ell]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} \\
&\text{ if and only if } (\mathfrak{I}([t'_1]\tfrac{t_0\cdots t_r}{x_0\cdots x_r}),\ldots,\mathfrak{I}([t'_\ell]\tfrac{t_0\cdots t_r}{x_0\cdots x_r})) \in \mathfrak{a}(R) \\
&\text{ if and only if } (\mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r}(t'_1),\ldots,\mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r}(t'_\ell)) \in \mathfrak{a}(R) \\
&\text{ if and only if } \mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r} \models Rt'_1\cdots t'_\ell.
\end{aligned}$$

Now suppose $\phi = \neg\psi$ for a formula ψ, and that the result holds for ψ. Then

$$\begin{aligned}
\mathfrak{I} \models [\neg\psi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} &\text{ if and only if } \mathfrak{I} \models \neg[\psi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} \\
&\text{ if and only if not } \mathfrak{I} \models [\psi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} \\
&\text{ if and only if not } \mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r} \models \psi \\
&\text{ if and only if } \mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r} \models \neg\psi.
\end{aligned}$$

Next suppose $\phi = (\psi \vee \chi)$ for formulas ψ and χ, and that the result holds for ψ and χ. Then we see

$$\begin{aligned}
\mathfrak{I} \models [(\psi \vee \chi)]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} &\text{ if and only if } \mathfrak{I} \models ([\psi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} \vee [\chi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r}) \\
&\text{ if and only if } \mathfrak{I} \models [\psi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} \text{ or } \mathfrak{I} \models [\chi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} \\
&\text{ if and only if } \mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r} \models \psi \text{ of } \mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r} \models \chi \\
&\text{ if and only if } \mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r} \models (\psi \vee \chi).
\end{aligned}$$

Finally, suppose $\phi = \exists x\psi$ for a variable x and a formula ψ, and that the result holds for ψ. Let $x_{i_1},\ldots,x_{i_s}$ be the variables among $x_0,\ldots,x_r$ such that $x_i \in \text{free}(\exists x\psi)$ and $x_i \neq t_i$. And let u be as in the definition of substitution. Then we get

$$\begin{aligned}
\mathfrak{I} \models [\exists x\psi]\tfrac{t_0\cdots t_r}{x_0\cdots x_r} &\text{ if and only if } \mathfrak{I} \models \exists u[\psi]\tfrac{t_{i_1}\cdots t_{i_s}u}{x_{i_1}\cdots x_{i_s}x} \\
&\text{ if and only if there is an } a \in A \text{ such that } \mathfrak{I}\tfrac{a}{u} \models [\psi]\tfrac{t_{i_1}\cdots t_{i_s}u}{x_{i_1}\cdots x_{i_s}x} \\
&\text{ if and only if there is an } a \in A \text{ such that } \mathfrak{I}\tfrac{a}{u}\tfrac{\mathfrak{I}\frac{a}{u}(t_{i_1})\cdots\mathfrak{I}\frac{a}{u}(t_{i_s})\mathfrak{I}\frac{a}{u}(u)}{x_{i_1}\cdots x_{i_s}x} \models \psi \\
\text{(1)}\quad &\text{ if and only if there is an } a \in A \text{ such that } \mathfrak{I}\tfrac{a}{u}\tfrac{\mathfrak{I}(t_{i_1})\cdots\mathfrak{I}(t_{i_s})a}{x_{i_1}\cdots x_{i_s}x} \models \psi \\
\text{(2)}\quad &\text{ if and only if there is an } a \in A \text{ such that } \mathfrak{I}\tfrac{\mathfrak{I}(t_{i_1})\cdots\mathfrak{I}(t_{i_s})a}{x_{i_1}\cdots x_{i_s}x} \models \psi \\
&\text{ if and only if there is an } a \in A \text{ such that } \mathfrak{I}\tfrac{\mathfrak{I}(t_{i_1})\cdots\mathfrak{I}(t_{i_s})}{x_{i_1}\cdots x_{i_s}}\tfrac{a}{x} \models \psi \\
&\text{ if and only if } \mathfrak{I}\tfrac{\mathfrak{I}(t_{i_1})\cdots\mathfrak{I}(t_{i_s})}{x_{i_1}\cdots x_{i_s}} \models \exists x\psi \\
&\text{ if and only if } \mathfrak{I}\tfrac{\mathfrak{I}(t_0)\cdots\mathfrak{I}(t_r)}{x_0\cdots x_r} \models \exists x\psi.
\end{aligned}$$

Note that (1) and (2) follow from the coincidence lemma. This completes the proof of the substitution lemma. ∎

Proposition 52. *Let S be a first-order language, let $r \in \mathbb{N}$, let $x_0, \ldots, x_r$ be pairwise distinct variables, let $t_0, \ldots, t_r$ be S-terms, and let ϕ be an S-formula. For $n \in \mathbb{N}$, if $\mathrm{free}_S(\phi) \subseteq \{x_0, \ldots, x_r\}$ and $\mathrm{var}_S(t_i) \subseteq \{v_0, \ldots, v_{n-1}\}$ then $\phi\frac{t_0 \cdots t_r}{x_0 \cdots x_r} \in L^S_n$. In particular, if each t_i is a constant symbol then $\phi\frac{t_0 \cdots t_r}{x_0 \cdots x_r}$ is an S-sentence.*

Proof. Exercise (uses a lemma which we have not stated — see book). ∎

Definition 53. Let S be a first-order language. We define the *rank* of a formula, $\mathrm{rk}_S : L^S \to \mathbb{N}$, inductively as follows:

$\mathrm{rk}_S(\phi) := 0$ for atomic formula ϕ;

$\mathrm{rk}_S(\neg\phi) := \mathrm{rk}_S(\phi) + 1$;

$\mathrm{rk}_S((\phi \vee \psi)) := \mathrm{rk}_S(\phi) + \mathrm{rk}_S(\psi) + 1$;

$\mathrm{rk}_S(\exists x \phi) := \mathrm{rk}_S(\phi) + 1$.

4 A Sequent Calculus

4.1 Sequent Rules

4.2 Structural Rules and Connective Rules

4.3 Derivable Connective Rules

4.4 Quantifier and Equality Rules

4.5 Further Derivable Rules and Sequents

Definition 54. We define the rules of the sequent calculus $\mathfrak{S}$:

Antecedent Rule (Ant):

$$\frac{\phi_1 \cdots \phi_n \phi}{\psi_1 \cdots \psi_m \phi} \quad \text{if } \phi_1, \ldots, \phi_n \in \{\psi_1, \ldots, \psi_m\}$$

Assumption Rule (Assm):

$$\frac{}{\phi_1 \cdots \phi_n \phi} \quad \text{if } \phi \in \{\phi_1, \ldots, \phi_n\}$$

Proof by Cases (PC):

$$\frac{\begin{array}{c}\phi_1 \cdots \phi_n \psi \phi \\ \phi_1 \cdots \phi_n \neg\psi \phi\end{array}}{\phi_1 \cdots \phi_n \phi}$$

Contradiction (Ctr):

$$\frac{\begin{array}{c}\phi_1 \cdots \phi_n \neg\phi \psi \\ \phi_1 \cdots \phi_n \neg\phi \neg\psi\end{array}}{\phi_1 \cdots \phi_n \phi}$$

$\vee$-rule for the Antecedent ($\vee$A):

$$\frac{\begin{array}{c}\phi_1 \cdots \phi_n \phi \chi \\ \phi_1 \cdots \phi_n \psi \chi\end{array}}{\phi_1 \cdots \phi_n (\phi \vee \psi) \chi}$$

$\vee$-rules for the Succedent ($\vee$S):

$$\frac{\phi_1 \cdots \phi_n \phi}{\phi_1 \cdots \phi_n (\phi \vee \psi)} \qquad \frac{\phi_1 \cdots \phi_n \phi}{\phi_1 \cdots \phi_n (\psi \vee \phi)}$$

$\exists$-introduction in the Antecedent ($\exists$A):

$$\frac{\phi_1 \cdots \phi_n \phi \frac{y}{x} \psi}{\phi_1 \cdots \phi_n \exists x \phi \psi} \quad \text{if } y \notin \text{free}((\phi_1 \vee \cdots \vee \phi_n)) \cup \text{free}(\exists x \phi) \cup \text{free}(\psi)$$

$\exists$-introduction in the Succedent ($\exists$S):

$$\frac{\phi_1 \cdots \phi_n \phi \frac{t}{x}}{\phi_1 \cdots \phi_n \exists x \phi}$$

Reflexive rule for equality ($\equiv$):

$$\frac{}{t \equiv t}$$

Substitution rule for equality (Sub):

$$\frac{\phi_1 \cdots \phi_n \phi \frac{t}{x}}{\phi_1 \cdots \phi_n t \equiv t' \phi \frac{t'}{x}}$$

Definition 55. Let S be a first-order language, let $n \in \mathbb{N}$, and let $\phi_1, \ldots, \phi_n, \phi \in L^S$. We say $\phi_1 \cdots \phi_n \phi$ is a *sequent*, with *antecedent* $\phi_1 \cdots \phi_n$ and *succedent* ϕ. If there is a derivation of $\phi_1 \cdots \phi_n \phi$ in the sequent calculus $\mathfrak{S}$, then we say $\phi_1 \cdots \phi_n \phi$ is *derivable* and write $\vdash_S \phi_1 \cdots \phi_n \phi$.

Definition 56. Let S be a first-order language, let ϕ be an S-formula, and let Φ be a set of S-formulas. We say ϕ is *formally provable* (or *derivable*) from Φ, and write $\Phi \vdash_S \phi$, if there is a finite subset $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$ such that $\vdash_S \phi_1 \cdots \phi_n \phi$.

Definition 57. Let S be a first-order language, and let $\phi_1 \cdots \phi_n \phi$ be a sequent. We say $\phi_1 \cdots \phi_n \phi$ is *correct* if $\{\phi_1, \ldots, \phi_n\} \vDash \phi$.

Definition 58. Let S be a first-order language. We say a rule concerning the sequent calculus is correct if whenever it is applied to correct sequents, it produces a correct sequent.

4.6 Summary and Example

Theorem 59. [Soundness Theorem] *Let S be a first-order language, let $\phi \in L^S$, and let $\Phi \subseteq L^S$. If $\Phi \vdash_S \phi$ then $\Phi \vDash \phi$.*

Proof. Suppose $\Phi \vdash_S \phi$. Then there are $\phi_1, \ldots, \phi_n \in \Phi$ such that $\vdash_S \phi_1 \cdots \phi_n \phi$. Hence it suffices to prove that all the rules of the sequent calculus $\mathfrak{S}$ are correct. Indeed, if we have this, then we get $\{\phi_1, \ldots, \phi_n\} \vDash \phi$, and hence $\Phi \vDash \phi$.

To this end, we go through the rules of $\mathfrak{S}$:

(Ant) Suppose $\phi_1, \ldots, \phi_n, \phi \in L^S$, and assume that $\{\phi_1, \ldots, \phi_n\} \vDash \phi$. We wish to show that $\{\psi_1, \ldots, \psi_m\} \vDash \phi$, given that $\phi_1, \ldots, \phi_n \in \{\psi_1, \ldots, \psi_m\}$. So suppose $\mathfrak{I}$ is an S-interpretation and that $\mathfrak{I} \vDash \{\psi_1, \ldots, \psi_m\}$. Since $\{\phi_1, \ldots, \phi_n\} \subseteq \{\psi_1, \ldots, \psi_m\}$, we have $\mathfrak{I} \vDash \{\phi_1, \ldots, \phi_n\}$. Thus by assumption we see $\mathfrak{I} \vDash \phi$. As $\mathfrak{I}$ was arbitrary, we conclude $\{\psi_1, \ldots, \psi_m\} \vDash \phi$. Thus the rule (Ant) is correct.

In fact, most of the remaining cases are similarly straightforward (with the possible exception of ($\exists$A)), and will therefore be left as exercises seeing as the sequent calculus is not our primary object of concern in this course. ∎

Remark 60. The converse of the soundness theorem, known as the completeness theorem, also holds. We will work towards proving this in the next chapter.

4.7 Consistency

Definition 61. Let S be a first-order language and let $\Phi \subseteq L^S$. We say Φ is *consistent*, and write $\mathrm{Con}_S\,\Phi$, if there is no formula $\phi \in L^S$ such that $\Phi \vdash_S \phi$ and $\Phi \vdash_S \neg\phi$. We say Φ is *inconsistent*, and write $\mathrm{Inc}_S\,\Phi$, if there is a formula $\phi \in L^S$ such that $\Phi \vdash_S \phi$ and $\Phi \vdash_S \neg\phi$.

Lemma 62. *Let S be a first-order language and let $\Phi \subseteq L^S$. Then $\mathrm{Inc}_S\,\Phi$ if and only if we have $\Phi \vdash_S \phi$ for all $\phi \in L^S$.*

Proof. Suppose first that we have $\Phi \vdash_S \phi$ for all $\phi \in L^S$. Note $v_0 \equiv v_0 \in L^S$ and $\neg v_0 \equiv v_0 \in L^S$. Hence $\Phi \vdash_S v_0 \equiv v_0$ and $\Phi \vdash_S \neg v_0 \equiv v_0$. Thus $\mathrm{Inc}_S\,\Phi$.

Conversely, suppose $\mathrm{Inc}_S\,\Phi$, and let $\phi \in L^S$. Since Φ is inconsistent, there exists a formula $\psi \in L^S$ and $\phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m \in \Phi$ such that there is a derivation of $\phi_1 \cdots \phi_n \psi$ and a derivation of $\psi_1 \cdots \psi_m \neg\psi$. So we get the following derivation:

$$
\begin{array}{lll}
 & \vdots & \\
(m) & \phi_1 \cdots \phi_n \psi & \\
 & \vdots & \\
(m+n) & \psi_1 \cdots \psi_m \neg\psi & \\
(m+n+1) & \phi_1 \cdots \phi_n \psi_1 \cdots \psi_m \neg\phi\psi & \text{(Ant) on } m \\
(m+n+2) & \phi_1 \cdots \phi_n \psi_1 \cdots \psi_m \neg\phi\neg\psi & \text{(Ant) on } m+n \\
(m+n+3) & \phi_1 \cdots \phi_n \psi_1 \cdots \psi_m \phi & \text{(Ctr) on } m+n+1 \text{ and } m+n+2
\end{array}
$$

As $\{phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m\} \subseteq \Phi$, we have $\Phi \vdash_S \phi$, as required. ∎

Corollary 63. *Let S be a first-order language and let $\Phi \subseteq L^S$. Then $\mathrm{Cons}_S\,\Phi$ if and only if there is a formula $\phi \in L^S$ which is not derivable form Φ.*

Lemma 64. *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. Then $\mathrm{Cons}_S\,\Phi$ if and only if for all finite subsets $\Phi_0 \subseteq \Phi$ we have $\mathrm{Cons}_S\,\Phi_0$.*

Proof. Suppose first $\mathrm{Cons}_S\,\Phi$, and let $\Phi_0 \subseteq \Phi$ be finite. Assume, for a contradiction, that $\mathrm{Inc}_S\,\Phi_0$. Then there is an S-formula ϕ such that $\Phi_0 \vdash_S \phi$ and $\Phi_0 \vdash_S \neg\phi$. But since $\Phi_0 \subseteq \Phi$, this means $\Phi \vdash_S \phi$ and $\Phi \vdash_S \neg\phi$, contradicting $\mathrm{Cons}_S\,\Phi$. Hence $\mathrm{Cons}_S\,\Phi_0$.

Conversely, suppose that for all finite subsets $\Phi_0 \subseteq \Phi$ we have $\mathrm{Cons}_S\,\Phi_0$. Assume, for a contradiction, $\mathrm{Inc}_S\,\Phi$. Then there is some $\phi \in L^S$ such that $\Phi \vdash_S \phi$ and $\Phi \vdash_S \neg\phi$. So there exist $\psi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m \in \Phi$ such that $\vdash_S \phi_1 \cdots \phi_n \phi$ and $\vdash_S \psi_1 \cdots \psi_m \phi$. Let $\Phi_0 = \{\psi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m\}$. Then $\Phi_0 \vdash_S \phi$ and $\Phi_0 \vdash_S \neg\phi$. Therefore $\mathrm{Inc}_S\,\Phi_0$. But $\Phi_0 \subseteq \Phi$ is finite. Contradiction. Thus $\mathrm{Cons}_S\,\Phi$. ∎

Lemma 65. *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. If* $\mathrm{Sat}\,\Phi$ *then* $\mathrm{Cons}_S\,\Phi$.

Proof. Suppose $\mathrm{Sat}_S\,\Phi$. Assume, for a contradiction, $\mathrm{Inc}_S\,\Phi$. Then there is some S-formula $\phi \in L^S$ such that $\Phi \vdash_S \phi$ and $\Phi \vdash_S \neg\phi$. By the soundness theorem, we have $\Phi \vDash \phi$ and $\Phi \vDash \neg\phi$. As $\mathrm{Sat}_S\,\Phi$, there is some S-interpretation $\mathfrak{I}$ such that $\mathfrak{I} \vDash \Phi$. So we have $\mathfrak{I} \vDash \phi$ and $\mathfrak{I} \vDash \neg\phi$. But this is absurd. Thus $\mathrm{Cons}_S\,\Phi$. ∎

Lemma 66. *Let S be a first-order language, let $\Phi \subseteq L^S$ be a set of S-formulas, and let $\phi \in L^S$ be an S-formula.*

(a) $\Phi \vdash_S \phi$ *if and only if* $\mathrm{Inc}_S\,\Phi \cup \{\neg\phi\}$.

(b) $\Phi \vdash_S \neg\phi$ *if and only if* $\mathrm{Inc}_S\,\Phi \cup \{\phi\}$.

(c) *If* $\operatorname{Con}_S \Phi$ *then either* $\operatorname{Con}_S \Phi \cup \{\neg\phi\}$ *or* $\operatorname{Con}_S \Phi \cup \{\neg\phi\}$.

Proof. (a) Suppose $\Phi \vdash_S \phi$. Then $\Phi \cup \{\neg\phi\} \vdash_S \phi$ (Ant). But also $\Phi \cup \{\neg\phi\} \vdash_S \neg\phi$ (Assm). Hence $\operatorname{Inc}_S \Phi \cup \{\neg\phi\}$.

Conversely, suppose $\operatorname{Inc}_S \Phi \cup \{\neg\phi\}$. We saw this means $\Phi \cup \{\neg\phi\} \vdash_S \psi$ for any $\psi \in L^S$, so in particular, we have $\Phi \cup \{\neg\phi\} \vdash_S \phi$. Hence for some $\phi_1, \ldots, \phi_n \in \Phi$ we have a derivation, which we can add to:

$$\begin{array}{ll} \vdots & \\ \phi_1 \cdots \phi_n \neg\phi\phi & \\ \phi_1 \cdots \phi_n \phi\phi & \text{(Assm)} \\ \phi_1 \cdots \phi_n \phi & \text{(PC)} \end{array}$$

Thus $\Phi \vdash_S \phi$.

(b) This is left as an exercise; it is extremely similar to part (a).

(c) Suppose $\operatorname{Con}_S \Phi$. Assume, for a contradiction, $\operatorname{Inc}_S \Phi \cup \{\phi\}$ and $\operatorname{Inc}_S \Phi \cup \{\neg\phi\}$. By the above, this means $\Phi \vdash_S \phi$ and $\Phi \vdash_S \neg\phi$. Hence $\operatorname{Inc}_S \Phi$. Contradiction. Thus either $\operatorname{Con}_S \Phi \cup \{\phi\}$ and $\operatorname{Con}_S \Phi \cup \{\neg\phi\}$. ∎

REMARK **67.** In the next chapter, we wish to prove the completeness theorem. To do this, we will show that any consistent set of formulas is satisfiable. Why is this sufficient?

LEMMA **68.** *For* $n \in \mathbb{N}$, *let* S_n *be a first-order language and let* $\Phi_n \subseteq L^{S_n}$ *be a set of* S_n*-formulas. If for all* $n \in \mathbb{N}$ *we have* $S_n \subseteq S_{n+1}$, $\operatorname{Con}_{S_n} \Phi_n$, *and* $\Phi_n \subseteq \Phi_{n+1}$, *then* $\operatorname{Con}_{\bigcup_{n\in\mathbb{N}} S_n} \bigcup_{n\in\mathbb{N}} \Phi_n$.

Proof. Assume, for a contradiction, $\operatorname{Inc}_{\bigcup_{n\in\mathbb{N}} S_n} \bigcup_{n\in\mathbb{N}} \Phi_n$. As we saw, this means that there is a finite subset $\Psi \subseteq \bigcup_{N\in\mathbb{N}} \Phi_n$ such that $\operatorname{Inc}_{\bigcup_{N\in\mathbb{N}} S_n} \Psi$. Since the sets of formulas are nested, there is some $k \in \mathbb{N}$ such that $\Psi \subseteq \Phi_k$. Hence $\operatorname{Inc}_{\bigcup_{N\in\mathbb{N}} S_n} \Phi_k$. In particular, this means that $\Phi_k \vdash_{\bigcup_{n\in\mathbb{N}} S_n} v_0 \equiv v_0$ and $\Phi_k \vdash_{\bigcup_{n\in\mathbb{N}} S_n} \neg v_0 \equiv v_0$. So there are $\phi_1, \ldots, \phi_r, \psi_1, \ldots, \psi_t \in \Phi_k$ such that $\vdash_{\bigcup_{n\in\mathbb{N}} S_n} \phi_1 \cdots \phi_r v_0 \equiv v_0$ and $\vdash_{\bigcup_{n\in\mathbb{N}} S_n} \psi_1 \cdots \psi_t \neg v_0 \equiv v_0$. Note that there are finitely many symbols in these derivations. As the languages are nested, there is some $m \geq k$ such that all the symbols are contained in $\mathbb{A}_{S_m}$. Hence $\vdash_{S_m} \phi_1 \cdots \phi_r v_0 \equiv v_0$ and $\vdash_{S_m} \psi_1 \cdots \psi_t \neg v_0 \equiv v_0$. Thus $\Phi_k \vdash_{S_m} v_0 \equiv v_0$ and $\Phi_k \vdash_{S_m} \neg v_0 \equiv v_0$. Therefore $\operatorname{Inc}_{S_m} \Phi_k$. But $\Phi_k \subseteq \Phi_m$, so we have $\operatorname{Inc}_{S_m} \Phi_m$. Contradiction. Thus the result holds. ∎

5 The Completeness Theorem

5.1 Henkin's Theorem

Definition 69. Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. We say Φ is *negation complete* if for every S-formula $\phi \in L^S$, either $\Phi \vdash_S \phi$ or $\Phi \vdash_S \neg\phi$.

Definition 70. Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. We say Φ *contains witnesses* if for all S-formulas $\phi \in L^S$ and all variables x, there is an S-term $t \in T^S$ such that $\Phi \vdash_S (\exists x\phi \to \phi\frac{t}{x})$.

Lemma 71. *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. If Φ is consistent, negation complete, and contains witnesses, then:*

(a) For all $\phi \in L^S$, $\Phi \vdash_S \neg\phi$ if and only if it is not the case that $\Phi \vdash_S \phi$;

(b) For all $\phi, \psi \in L^S$, $\Phi \vdash_S (\phi \vee \psi)$ if and only if either $\Phi \vdash_S \phi$ or $\Phi \vdash_S \psi$;

(c) For all variables x and $\phi \in L^S$, $\Phi \vdash_S \exists x\phi$ if and only if there is a term $t \in T^S$ with $\Phi \vdash_S \phi\frac{t}{x}$.

Proof. (a) Suppose $\Phi \vdash_S \neg\phi$. Since $\mathrm{Con}_S\,\Phi$, we cannot have $\Phi \vdash_S \phi$.

Conversely, suppose we do not have $\Phi \vdash_S \phi$. Since Φ is negation complete, we must have $\Phi \vdash_S \neg\phi$.

(b) Suppose $\Phi \vdash_S (\phi \vee \psi)$. Assume we do not have $\Phi \vdash_S \phi$. Since Φ is negation complete, we must have $\Phi \vdash_S \neg\phi$. Note that one can derive the rule

$$\begin{array}{c} \phi_1 \cdots \phi_n (\phi \vee \psi) \\ \underline{\phi_1 \cdots \phi_n \neg\phi} \\ \phi_1 \cdots \phi_n \psi \end{array}$$

(see page 65). We therefore have $\Phi \vdash_S \psi$, as required.

Suppose now $\Phi \vdash_S \phi$. Then $\Phi \vdash_S (\phi \vee \psi)$ (by ($\vee$S)).

Suppose now $\Phi \vdash_S \psi$. Then $\Phi \vdash_S (\phi \vee \psi)$ (by ($\vee$S)).

(c) Suppose $\Phi \vdash_S \exists x\phi$. Since Φ contains witnesses, there is some term $t \in T^S$ such that $\Phi \vdash_S (\exists x\phi \to \phi\frac{t}{x})$. By modus ponens (see page 65), we get $\Phi \vdash_S \phi\frac{t}{x}$.

Conversely, suppose there is some term $t \in L^S$ such that $\Phi \vdash_S \phi\frac{t}{x}$. Then $\Phi \vdash_S \exists x\phi$ (by ($\exists$S)).

∎

REMARK **72.** As we said, we wish to prove the completeness theorem, and to do this, we will show that any consistent set of formulas is satisfiable. So given a consistent set of formulas, how are we going to construct an interpretation which models this set? About the only thing we can try to do is make our universe be the set of terms, and have everything map to 'itself', in some natural sense. But what if $\Phi \vdash_S fx \equiv fy$ (for distinct variables x and y)? Then we would need $\mathfrak{I}(fx) = \mathfrak{I}(fy)$. But $\mathfrak{I}(fx) = fx$ and $\mathfrak{I}(fy) = fy$, which are not equal.

It turns out that this is the only issue which arises, and that we can look at the set of terms modulo an equivalence relation which will make everything work out.

DEFINITION **73.** Let S be a first-order language and let $t_1 \in T^S$ and $t_2 \in T^S$ be S-terms. We write $t_1 \sim_S^\Phi t_2$ if $\Phi \vdash_S t_1 \equiv t_2$.

LEMMA **74.** *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. Then $\sim_S^\Phi$ is an equivalence relation on T^S.*

Proof. First we show that $\sim_S^\Phi$ is reflexive. Let $t \in t^S$. Then $\Phi \vdash_S t \equiv t$ (by ($\equiv$)). Hence $t \sim_S^\Phi t$.

Symmetry and transitivity are left as exercises (use derived rules — see book). ∎

LEMMA **75.** *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. Let $t_1, \ldots, t_n, t_1', \ldots, t_n' \in T^S$ be S-terms, let $f \in S$ be an n-ary function symbol, and let $R \in S$ be an n-ary function symbol. If $t_1 \sim_S^\Phi t_1', \ldots, t_n \sim_S^\Phi t_n'$, then $ft_1 \cdots t_n \sim_S^\Phi ft_1' \cdots t_n'$, and $\Phi \vdash_S Rt_1 \cdots t_n$ if and only if $\Phi \vdash_S Rt_1' \cdots t_n'$.*

Proof. Exercise (see book). ∎

DEFINITION **76.** Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. For $t \in T^S$, we let $\overline{t}_S^\Phi := \{t' \in T^S; t \sim_S^\Phi t'\}$ denote the equivalence class of t in T^S. We let $T_S^\Phi := T^S / \sim_S^\Phi = \{\overline{t}_S^\Phi; t \in T^S\}$ denote T^S modulo the equivalence relation $\sim_S^\Phi$.

DEFINITION **77.** Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. Define $\mathfrak{a}_S^\Phi$ as follows:

For n-ary relation symbol $R \in S$,

$$\mathfrak{a}_S^\Phi(R) := \{(\overline{t_1}_S^\Phi, \ldots, \overline{t_n}_S^\Phi) \in T_S^\Phi; \Phi \vdash_S Rt_1 \cdots t_n\};$$

For n-ary function symbol $f \in S$,

$$\mathfrak{a}_S^\Phi(f)((\overline{t_1}_S^\Phi, \ldots, \overline{t_n}_S^\Phi)) := \overline{ft_1 \cdots t_n}_S^\Phi;$$

For constant symbol $c \in S$,

$$\mathfrak{a}_S^\Phi(c) := \overline{c}_S^\Phi.$$

We set $\mathfrak{T}_S^\Phi := (T_S^\Phi, \mathfrak{a}_S^\Phi)$, an S-structure.

We also let $\beta_S^\Phi(v_i) := \overline{v_i}_S^\Phi$ for each $i \in \mathbb{N}$, and define the *term interpretation* associated with Φ to be $(\mathfrak{T}_S^\Phi, \beta_S^\Phi)$.

Remark 78. By the previous lemma, the definitions above are independent of which representatives we pick, and are hence well-defined.

Theorem 79. *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. If Φ is consistent, negation complete, and contains witnesses, then Φ is satisfiable.*

Proof. We will show that $\mathfrak{I}_S^\Phi \vDash \Phi$, which certainly shows $\operatorname{Sat}\Phi$.

First we show that for all S-terms $t \in T^S$ we have $\mathfrak{I}_S^\Phi(t) = \overline{t}_S^\Phi$. We proceed by induction on S-terms. Let x be a variable. Then $\mathfrak{I}_S^\Phi(x) = \beta_S^\Phi(x) = \overline{x}_S^\Phi$. Let $c \in S$ be a constant symbol. Then $\mathfrak{I}_S^\Phi(c) = \mathfrak{a}_S^\Phi(c) = \overline{c}_S^\Phi$. Suppose $f \in S$ is an n-ary function symbol and $t_1, \ldots, t_n \in T^S$ are S-terms for which the result holds. Then

$$\mathfrak{I}_S^\Phi(ft_1\cdots t_n) = \mathfrak{a}_S^\Phi(f)((\mathfrak{I}_S^\Phi(t_1), \ldots, \mathfrak{I}_S^\Phi(t_n))) = \mathfrak{a}_S^\Phi(f)((\overline{t_1}_S^\Phi, \ldots, \overline{t_n}_S^\Phi)) = \overline{ft_1\cdots t_n}_S^\Phi.$$

Next, we show that for all S-formulas $\phi \in L^S$, we have $\mathfrak{I}_S^\Phi \vDash \phi$ if and only if $\Phi \vdash_S \phi$. We proceed by induction on the rank of $\operatorname{rk}_S(\phi)$. Suppose $\operatorname{rk}_S(\phi) = 0$. If t_1 and t_2 are terms with $\phi = t_1 \equiv t_2$, then

$$\begin{aligned}
\mathfrak{I}_S^\Phi \vDash t_1 \equiv t_2 &\text{ if and only if } \mathfrak{I}_S^\Phi(t_1) = \mathfrak{I}_S^\Phi(t_2) \\
&\text{ if and only if } \overline{t_1}_S^\Phi = \overline{t_2}_S^\Phi \\
&\text{ if and only if } t_1 \sim_S^\Phi t_2 \\
&\text{ if and only if } \Phi \vdash_S t_1 \equiv t_2.
\end{aligned}$$

Otherwise, we must have $\phi = Rt_1\cdots t_n$ for an n-ary relation symbol $R \in S$ and terms $t_1, \ldots, t_n \in T^S$. In this case

$$\begin{aligned}
\mathfrak{I}_S^\Phi \vDash Rt_1\cdots t_n &\text{ if and only if } (\mathfrak{I}_S^\Phi(t_1), \ldots, \mathfrak{I}_S^\Phi(t_n)) \in \mathfrak{a}_S^\Phi(R) \\
&\text{ if and only if } (\overline{t_1}_S^\Phi, \ldots, \overline{t_n}_S^\Phi) \in \mathfrak{a}_S^\Phi(R) \\
&\text{ if and only if } \Phi \vdash_S Rt_1\cdots t_n.
\end{aligned}$$

Now suppose $\operatorname{rk}_S(\phi) > 0$, and that the result holds for all formulas of lesser rank. This means ϕ is non-atomic. First suppose $\phi = \neg\psi$ for a formula $\psi \in L^S$. Note that $\operatorname{rk}_S(\psi) < \operatorname{rk}_S(\neg\psi)$. Hence

$$\begin{aligned}
\mathfrak{I}_S^\Phi \vDash \neg\phi &\text{ if and only if we do not have } \mathfrak{I}_S^\Phi \vDash \phi \\
&\text{ if and only if we do not have } \Phi \vdash_S \phi \\
&\text{ if and only if } \Phi \vdash_S \neg\phi.
\end{aligned}$$

Note the final step holds by lemma (71). Now suppose $\phi = (\chi \vee \psi)$ for formulas $\chi, \psi \in L^S$. Note that $\mathrm{rk}_S(\chi), \mathrm{rk}_S(\psi) < \mathrm{rk}_S((\phi \vee \psi))$. Hence

$$\begin{aligned}\mathfrak{I}_S^\Phi \models (\phi \vee \psi) &\text{ if and only if } \mathfrak{I}_S^\Phi \models \phi \text{ or } \mathfrak{I}_S^\Phi \models \psi\\ &\text{ if and only if } \Phi \vdash_S \phi \text{ or } \Phi \vdash_S \psi\\ &\text{ if and only if } \Phi \vdash_S (\phi \vee \psi).\end{aligned}$$

Note again that the final step holds by lemma (71). Finally, suppose $\phi = \exists x \psi$ for a formula $\psi \in L^S$ and a variable x. Note that $\mathrm{rk}_S(\psi\frac{t}{x}) < \mathrm{rk}_S(\exists x \psi)$ for any S-term t. Hence

$$\begin{aligned}\mathfrak{I}_S^\Phi \models \exists x \psi &\text{ if and only if there is some } a \in T_S^\Phi \text{ such that } \mathfrak{I}_S^\Phi \tfrac{a}{x} \models \psi\\ &\text{ if and only if there is some } t \in T^S \text{ such that } \mathfrak{I}_S^\Phi \tfrac{\bar{t}_S^\Phi}{x} \models \psi\\ &\text{ if and only if there is some } t \in T^S \text{ such that } \mathfrak{I}_S^\Phi \tfrac{\mathfrak{I}_S^\Phi(t)}{x} \models \psi\\ &\text{ if and only if there is some } t \in T^S \text{ such that } \mathfrak{I}_S^\Phi \models \psi\tfrac{t}{x}\\ &\text{ if and only if there is some } t \in T^S \text{ such that } \Phi \vdash_S \psi\tfrac{t}{x}\\ &\text{ if and only if } \Phi \vdash_S \exists x \psi.\end{aligned}$$

Once again, note that the final step holds by lemma (71).

Finally, we show $\mathfrak{I}_S^\Phi \models \Phi$. Let $\phi \in \Phi$. Then certainly $\Phi \vdash_S \phi$. Therefore $\mathfrak{I}_S^\Phi \models \phi$. ■

5.2 Satisfiability of Consistent Sets of Formulas (the Countable Case)

LEMMA 80. *Let S be a **countable** first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. If $\mathrm{Con}_S\,\Phi$ and $\mathrm{free}_S(\Phi)$ is finite, then there is some $\Theta \subseteq L^S$ such that $\Phi \subseteq \Theta$ and Θ is consistent, negation complete, and contains witnesses.*

Proof. First we construct a $\Psi \subseteq L^S$ which contains witnesses and satisfies $\Phi \subseteq \Psi$. As S is countable, so is L^S and any subset of it. Let $\{\exists x_n \phi_n;\, n \in \mathbb{N}\}$ be an enumeration of the S-formulas which start with an existential quantifier. We inductively define ψ_n for $n \in \mathbb{N}$. Let $n \in \mathbb{N}$ and suppose we have defined ψ_m for all $m < n$. Let $y_n \notin \mathrm{free}_S(\Phi \cup \{\psi_m;\, m < n\} \cup \{\exists x_n \phi_n\})$ have least index (this exists because $\mathrm{free}_S(\Phi)$ is finite and there are finitely many symbols in $\{\phi_m;\, m < n\} \cup \{\exists x_n \phi_n\}$). Define

$$\psi_n := (\exists x_n \phi_n \to \phi_n \tfrac{y_n}{x_n})$$

and let $\Psi = \Phi \cup \{\psi_n;\, n \in \mathbb{N}\}$.

Certainly $\Phi \subseteq \Psi \subseteq L^S$. We claim that Ψ is consistent and contains witnesses. To see that Ψ contains witnesses, suppose x is a variable and $\phi \in L^S$. Then $\exists x \phi = \exists x_n \phi_n$ for some $n \in \mathbb{N}$ by construction. Consider the S-term y_n. Note $\Psi \vdash_S (\exists x \phi \to \phi\frac{y_n}{x})$ because $(\exists x \phi \to \phi\frac{y_n}{x}) = \psi_n \in \Psi$. Hence Ψ contains witnesses.

Next we show that Ψ is consistent. Let $\Phi_n = \Phi \cup \{\psi_m;\ m < n\}$. Since $\Psi = \bigcup_{N\in\mathbb{N}} \Phi_n$ with $\Phi_n \subseteq \Phi_{n+1}$ for all $n \in \mathbb{N}$, it suffices to show $\mathrm{Cons}_S\,\Phi_n$ for all $n \in \mathbb{N}$ (by lemma (68)). We proceed by induction on n. Note $\Phi_0 = \Phi$ so $\mathrm{Cons}_S\,\Phi_0$ because $\mathrm{Cons}_S\,\Phi$. Now let $n \in \mathbb{N}$ and suppose $\mathrm{Cons}_S\,\Phi_n$. Assume, for a contradiction, $\mathrm{Inc}_S\,\Phi_{n+1}$. Note that $\Phi_{n+1} = \Phi_n \cup \{\psi_n\}$. Suppose $\phi \in L_0^S$ is an S-sentence. Since Φ_{n+1} is inconsistent, there exist $\chi_1, \ldots, \chi_\ell \in \Phi_n$ such that $\vdash_S \chi_1 \ldots, \chi_\ell \psi_n \phi$. As an exercise, derive the rules

$$\frac{\phi_1 \cdots \phi_n (\psi \vee \chi)\phi}{\phi_1 \cdots \phi_n \psi \phi} \quad \text{and} \quad \frac{\phi_1 \cdots \phi_n (\psi \vee \chi)\phi}{\phi_1 \cdots \phi_n \chi \phi}.$$

As we said, we have a derivation of $\chi_1 \cdots \chi_\ell \psi_n \phi$, which we can add to now:

$$\begin{array}{lll}
 & \vdots & \\
(m) & \chi_1 \cdots \chi_\ell (\neg \exists x_n \phi_n \vee \phi_n \frac{y_n}{x_n})\phi & \\
(m+1) & \chi_1 \cdots \chi_\ell \neg \exists x_n \phi_n \phi & \text{(exercise on } m) \\
(m+2) & \chi_1 \cdots \chi_\ell \phi_n \frac{y_n}{x_n} \phi & \text{(exercise on } m) \\
(m+3) & \chi_1 \cdots \chi_\ell \exists x_n \phi_n \phi & ((\exists A) \text{ on } m+2 \text{ since } y_n \notin \mathrm{free}(\{\chi_1, \ldots, \chi_\ell\} \cup \{\exists x_n \phi_n\} \cup \{\phi\})) \\
(m+4) & \chi_1 \cdots \chi_\ell \phi & ((\mathrm{PC}) \text{ on } m+1 \text{ and } m+2)
\end{array}$$

In particular, if we take $\phi = \exists v_0 v_0 \equiv v_0$ we get $\Phi_n \vdash_S \exists v_0 v_0 \equiv v_0$, and if we take $\phi = \neg \exists v_0 v_0 \equiv v_0$ we get $\Phi_n \vdash_S \neg \exists v_0 v_0 \equiv v_0$ (because $\chi_1, \ldots, \chi_\ell \in \Phi_n$). Therefore $\mathrm{Inc}_S\,\Phi_n$. Contradiction. Thus $\mathrm{Cons}_S\,\Phi_{n+1}$. This shows $\mathrm{Cons}_S\,\Psi$, and completes the verification of the claim.

Now that we have constructed Ψ, we will construct $\Theta \subseteq L^S$ which is consistent, negation complete, and contains witnesses, with $\Psi \subseteq \Theta$. This will complete the proof of the lemma. As L^S is countable, let $L^S = \{\phi_n;\ n \in \mathbb{N}\}$ be an enumeration. Define $\Theta_0 = \Psi$ and for $n \in \mathbb{N}$

$$\Theta_{n+1} := \begin{cases} \Theta_n \cup \{\phi_n\} & \text{if } \mathrm{Cons}_S\,\Theta_n \cup \{\phi_n\} \\ \Theta_n & \text{if } \mathrm{Inc}_S\,\Theta_n \cup \{\phi_n\}. \end{cases}$$

Let $\Theta = \bigcup_{n\in\mathbb{N}} \Theta_n$.

Certainly $\Phi \subseteq \Psi \subseteq \Theta \subseteq L^S$. We claim that Θ is consistent, negation complete, and contains witnesses. That Θ contains witnesses follows from the fact that Ψ contains witnesses. Indeed, if $\Psi \vdash_S (\exists x \phi \to \phi \frac{t}{x})$ then $\Theta \vdash_S (\exists x \phi \to \phi \frac{t}{x})$ because $\Psi \subseteq \Theta$. To show $\mathrm{Cons}_S\,\Theta$, it again suffices to show $\mathrm{Cons}_S\,\Theta_n$ for all $n \in \mathbb{N}$. We proceed by induction on n. Since $\Theta_0 = \Psi$ and we have $\mathrm{Cons}_S\,\Phi$, we get $\mathrm{Cons}_S\,\Theta_0$. Let $n \in \mathbb{N}$ and suppose $\mathrm{Cons}_S\,\Theta_n$. If $\mathrm{Cons}_S\,\Theta_n \cup \{\phi_n\}$ then $\Theta_{n+1} = \Theta_n \cup \{\phi_n\}$ by definition, and hence $\mathrm{Cons}_S\,\Theta_{n+1}$. Otherwise, $\Theta_{n+1} = \Theta_n$ so by inductive assumption $\mathrm{Cons}_S\,\Theta_{n+1}$. Thus $\mathrm{Cons}_S\,\Theta$. Finally, we need only show that Θ is negation complete. Let $\phi \in L^S$. Then $\phi = \phi_n$ for some $n \in \mathbb{N}$. Assume we do not have $\Theta \vdash_S \neg \phi_n$. By a lemma in the previous chapter, we have $\mathrm{Cons}_S\,\Theta \cup \{\phi_n\}$. But $\Theta_n \subseteq \Theta$, so this means $\mathrm{Cons}_S\,\Theta_n \cup \{\phi_n\}$. Hence $\phi_n \in \Theta_{n+1} \subseteq \Theta$. Therefore $\Theta \vdash_S \phi_n$ and so Θ is negation complete. ∎

5.3 Satisfiability of Consistent Sets of Formulas (the General Case)

Definition 81. Let S be a first-order language and let $\phi \in L^S$ be an S-formula. Set c_ϕ^S to be a distinct constant symbol which is not already in S. Define

$$S^* := S \cup \left\{c^S_{\exists x\phi};\ \exists x\phi \in L^S\right\}$$

and let

$$W(S) := \left\{(\exists x\phi \to \phi\frac{c^S_{\exists x\phi}}{x});\ \exists x\phi \in L^S\right\}.$$

Lemma 82. *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. If* $\mathrm{Con}_S\,\Phi$ *then* $\mathrm{Con}_{S^*}\,\Phi \cup W(S)$.

Proof. Recall $\mathrm{Con}_{S^*}\,\Phi \cup W(S)$ if and only if $\mathrm{Con}_{S^*}\,\Phi_0^*$ for all finite subsets $\Phi_0^* \subseteq \Phi \cup W(S)$. So let $\Phi_0^* \subseteq \Phi \cup W(S)$ be finite. We will show $\mathrm{Sat}_{S^*}\,\Phi_0^*$, since this implies $\mathrm{Con}_{S^*}\,\Phi_0^*$. Write

$$\Phi_0^* = \Phi_0 \cup \{(\exists x_1\phi_1 \to \phi_1\frac{c_{\exists x_1\phi_1}}{x_1}), \dots, (\exists x_n\phi_n \to \phi_n\frac{c_{\exists x_n\phi_n}}{x_n})\},$$

where $\Phi_0 \subseteq \Phi$ is finite.

Since Φ_0 is finite, there is some countable language $S_0 \subseteq S$ such that $\Phi_0 \subseteq L^{S_0}$. Clearly $\mathrm{Con}_S\,\Phi$ implies $\mathrm{Con}_{S_0}\,\Phi$. And since $\Phi_0 \subseteq \Phi$, we get $\mathrm{Con}_{S_0}\,\Phi_0$. Moreover, $\mathrm{free}_S(\Phi_0)$ is finite (because there are finitely many symbols in Φ_0). Hence by the previous lemma, we may conclude that there is some $\Theta \subseteq L^{S_0}$ which is consistent, negation complete, and contains witnesses, and such that $\Phi_0 \subseteq \Theta$. By the theorem before that, we conclude $\mathrm{Sat}_{S_0}\,\Phi_0$.

So let $\mathfrak{I}' = ((A, \mathfrak{a}'), \beta)$ be an S_0-interpretation of Φ_0 with $\mathfrak{I}' \vDash \Phi_0$. Extend $\mathfrak{a}'$ to $\mathfrak{a}$ defined on all of S. By the coincidence lemma, $\mathfrak{I} = ((A, \mathfrak{a}), \beta)$ is an S-interpretation with $\mathfrak{I} \vDash \Phi_0$. We need only extend this to an S^*-interpretation $\mathfrak{I}^*$ such that $\mathfrak{I}^* \vDash \Phi_0^*$.

Fix $b \in A$. Pick, for $i \in \{1, \dots, n\}$, $a_i \in A$ such that if $\mathfrak{I} \vDash \exists x_i\phi_i$ then $\mathfrak{I}\frac{a_i}{x_i}$, and otherwise just set $a_i = b$. Extend $(A, \mathfrak{a})$ to an S^*-structure by

$$\mathfrak{a}^*(c_{\exists x_i\phi_i}) = a_i$$

for $i \in \{1, \dots, n\}$, and

$$\mathfrak{a}^*(c_\phi) = b$$

otherwise. Then $\mathfrak{I}^* = ((A, \mathfrak{a}^*), \beta)$ is an S^*-interpretation.

We claim $\mathfrak{I}^* \vDash \Phi_0^*$. By the coincidence lemma, $\mathfrak{I}^* \vDash \Phi_0$. So it remains to show $\mathfrak{I}^* \vDash (\exists x_i\phi_i \to \phi_i\frac{c_{\exists x_i\phi_i}}{x_i})$ for each $i \in \{1, \dots, n\}$. To this end, let $i \in \{1, \dots, n\}$. Suppose $\mathfrak{I}^* \vDash \exists x_i\phi_i$. By the coincidence lemma, this means $\mathfrak{I} \vDash \exists x_i\phi_i$. Hence $\mathfrak{I}\frac{a_i}{x_i} \vDash \phi_i$. By the coincidence lemma again, we have $\mathfrak{I}^*\frac{a_i}{x_i} \vDash \phi_i$. By definition of $\mathfrak{a}^*$, this is just $\mathfrak{I}^*\frac{\mathfrak{I}^*(c_{\exists x_i\phi_i})}{x_i} \vDash \phi_i$. By the substitution lemma, we get $\mathfrak{I}^* \vDash \phi_i\frac{c_{\exists x_i\phi_i}}{x_i}$. Therefore $\mathfrak{I}^* \vDash (\exists x_i\phi_i \to \phi_i\frac{c_{\exists x_i\phi_i}}{x_i})$.

So we have $\mathfrak{I}^* \vDash \Phi_0^*$. This means $\mathrm{Sat}\,\Phi_0^*$. We saw that this implies $\mathrm{Con}_{S^*}\,\Phi_0^*$. As Φ_0^* was an arbitrary finite subset, we have $\mathrm{Con}_{S^*}\,\Phi \cup W(S)$. ∎

THEOREM **83.** *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. If* $\mathrm{Con}_S \Phi$ *then* $\mathrm{Sat}\,\Phi$.

Proof. First we extend S to a bigger language S' and Φ to a bigger set Ψ of S'-formulas, such that $\mathrm{Con}_{S'}$ and such that Ψ contains witnesses in S'. To this end, let $S_0 = S$, and recursively define $S_{n+1} = {S_n}^*$ for $n \in \mathbb{N}$. Also let $\Phi_0 = \Phi$, and recursively define $\Phi_{n+1} = \Phi_n \cup \mathcal{W}(S_n)$. Note that $S_0 \subseteq S_1 \subseteq \cdots$ and $\Phi_0 \subseteq \Phi_1 \subseteq \cdots$. Set $S' = \bigcup_{n \in \mathbb{N}} S_n$ and $\Psi = \bigcup_{n \in \mathbb{N}} \Phi_n$. A simple induction, using the previous lemma, shows that for each $n \in \mathbb{N}$ we have $\mathrm{Con}_{S_n} \Phi_n$. By lemma (68) we conclude $\mathrm{Con}_{S'} \Psi$. Moreover, Ψ contains witnesses in S', because if $\exists x\phi \in L^{S'}$, then $\exists x\phi \in L^{S_n}$ for some $n \in \mathbb{N}$, and so $(\exists x\phi \to \phi\frac{c_{\exists x\phi}}{x}) \in \mathcal{W}(S_n) \subseteq \Psi$.

Next we extend Ψ to a bigger set Θ of S'-formulas which is negation complete. Set

$$\mathcal{U} = \{\Gamma; \Psi \subseteq \Gamma \subseteq L^{S'} \text{ and } \mathrm{Con}_{S'} \Gamma\}.$$

We can partially order $\mathcal{U}$ by inclusion (that is, by $\subseteq$). Note that $\mathcal{U} \neq \varnothing$ since $\Psi \in \mathcal{U}$. Thus we may apply Zorn's lemma.

Suppose $\mathcal{D} \subseteq \mathcal{U}$ is a chain. Take $\Theta_1 = \bigcup \mathcal{D}$. Certainly $\Psi \subseteq \Theta_1 \subseteq L^{S'}$. To show that $\Theta_1 \in \mathcal{U}$, it remains to show $\mathrm{Con}_{S'} \Theta_1$. It suffices to show $\mathrm{Con}_{S'} \Theta_0$ for any finite subset $\Theta_0 \subseteq \Theta_1$, so let $\Theta_0 \subseteq \Theta_1$ be finite. As Θ_0 is finite and $\mathcal{D}$ is a chain, we have $\Theta_0 \subseteq \Gamma$ for some $\Gamma \in \mathcal{D}$. So we see that $\mathrm{Con}_{S'} \Gamma$ (as $\Gamma \in \mathcal{U}$) implies $\mathrm{Con}_{S'} \Theta_0$. Hence $\mathrm{Con}_{S'} \Theta_1$. Thus $\Theta_1 \in \mathcal{U}$, and it is clearly an upper bound for $\mathcal{D}$.

By Zorn's lemma, there exists a maximal element $\Theta \in \mathcal{U}$. Note that we immediately know that $\Psi \subseteq \Theta$ and that $\mathrm{Con}_{S'} \Theta$. We need only show that Θ is negation complete. So suppose $\phi \in L^{S'}$. Since $\mathrm{Con}_{S'} \Theta$, we have either $\mathrm{Con}_{S'} \Theta \cup \{\phi\}$ or $\mathrm{Con}_{S'} \Theta \cup \{\neg\phi\}$ (by lemma (66)). But since Θ is maximal, this forces us to have either $\Theta \cup \{\phi\} = \Theta$ or $\Theta \cup \{\neg\phi\}$. That is to say, either $\phi \in \Theta$ or $\neg\phi \in \Theta$. But now it clearly is the case that $\Theta \vdash_{S'} \phi$ or $\Theta \vdash_{S'} \neg\phi$. Thus Θ is negation complete (and consistent).

Furthermore, Θ contains witnesses because Ψ did and $\Psi \subseteq \Theta$. Thus Θ is a consistent, negation complete set which contains witnesses. By theorem (79), $\mathrm{Sat}\,\Theta$, as required. ∎

5.4 The Completeness Theorem

THEOREM **84.** [COMPLETENESS THEOREM] *Let S be a first-order language, let $\Phi \subseteq L^S$ be a set of S-formulas, and let $\phi \in L^S$ be an S-formula. If $\Phi \vDash \phi$ then $\Phi \vdash_S \phi$.*

Proof. Suppose $\Phi \vDash \phi$. Assume, for a contradiction, that we do not have $\Phi \vdash_S \phi$. Consider $\Phi \cup \{\neg\phi\}$. Since $\Phi \vDash \phi$, this set is not satisfiable. But it is consistent because $\Phi \nvdash_S \phi$ (see lemma (66)). So we have a consistent set which is not satisfiable. This contradicts the previous theorem. Thus $\Phi \vdash_S \phi$. ∎

COROLLARY **85.** *Let S_1 and S_2 be a first-order language, let $\phi \in L^{S_1 \cap S_2}$, and let $\Phi \subseteq L^{S_1 \cap S_2}$. Then $\Phi \vdash_{S_1} \phi$ if and only if $\Phi \vdash_{S_2} \phi$, and* $\mathrm{Con}_{S_1} \Phi$ *if and only if* $\mathrm{Con}_{S_2} \Phi$.

Proof. By (36) and the completeness and soundness theorems, we have $\Phi \vdash_{S_1} \phi$ if and only if $\Phi \vDash_{S_1} \phi$ if and only if $\Phi \vDash_{S_2} \phi$ if and only if $\Phi \vdash_{S_2} \phi$.

By (41) and the completeness and soundness theorems, we have $\mathrm{Con}_{S_1} \Phi$ if and only if $\mathrm{Sat}_{S_1} \Phi$ if and only if $\mathrm{Sat}_{S_2} \Phi$ if and only if $\mathrm{Con}_{S_2} \Phi$. ∎

6 The Löwenheim-Skolem Theorem and the Compactness Theorem

6.1 The Löwenheim-Skolem Theorem

THEOREM 86. [LÖWENHEIM-SKOLEM THEOREM] *Let S be a first-order language and let $\Phi \subseteq L^S$ be a countable set of S-formulas. If* $\mathrm{Sat}\,\Phi$ *then there is an S-interpretation with a countable domain which models Φ.*

Proof. As Φ is countable, countably many symbols appear in it, and so we can pick a countable language $S_0 \subseteq S$ which $\Phi \subseteq L^{S_0}$. In theorem (83), the extension Θ we get is countable, because S_0 is countable. Indeed, if S_0 is countable then ${S_0}^*$ is countable, and hence the set S' in the theorem is countable. It follows that $L^{S'}$ is countable. As $\Theta \subseteq L^{S'}$, Θ is countable.

Finally, note that the domain of the term interpretation is $L^{S'}$ modded out by some equivalence relation. In particular, this domain is countable, as required. ∎

6.2 The Compactness Theorem

THEOREM 87. [COMPACTNESS THEOREM] *Let S be a first-order language, let $\phi \in L^S$ be an S-formula, and let $\Phi \subseteq L^S$ be a set of S-formulas.*

(a) Then $\Phi \vDash \phi$ if and only if there is a finite subset $\Phi_0 \subseteq \Phi$ such that $\Phi_0 \vDash \phi$.

(b) Then $\mathrm{Sat}\,\Phi$ *if and only if for all finite subsets $\Phi_0 \subseteq \Phi$ we have* $\mathrm{Sat}\,\Phi_0$.

Proof. (a) Suppose $\Phi \vDash \phi$. Then $\Phi \vdash \phi$. Hence there is a some $\{\phi_1, \dots, \phi_n\} \subseteq \Phi$ such that $\vdash \phi_1 \cdots \phi_n \phi$. Whence $\{\phi_1, \dots, \phi_n\} \vdash \phi$. Therefore $\{\phi_1, \dots, \phi_n\} \vDash \phi$.

Conversely, suppose there is a finite $\Phi_0 \subseteq \Phi$ such that $\Phi_0 \vDash \phi$. Clearly $\Phi \vDash \phi$.

(b) Suppose $\mathrm{Sat}\,\Phi$. So there is some S-interpretation $\mathfrak{I}$ such that $\mathfrak{I} \vDash \Phi$. If $\Phi_0 \subseteq \Phi$ then $\mathfrak{I} \vDash \Phi_0$. In particular, this holds for all finite subsets Φ_0. Hence $\mathrm{Sat}\,\Phi_0$ for all finite subsets $\Phi_0 \subseteq \Phi$.

Conversely, suppose for all finite subsets $\Phi_0 \subseteq \Phi$ we have $\mathrm{Sat}\,\Phi_0$. Assume, for a contradiction, we do not have $\mathrm{Sat}\,\Phi$. Then $\mathrm{Inc}\,\Phi$. Hence there is some finite subset $\Phi_0 \subseteq \Phi$ such that $\mathrm{Inc}\,\Phi_0$. Contradiction. Thus $\mathrm{Sat}\,\Phi$.

∎

THEOREM 88. *Let S be a first-order language and let $\Phi \subseteq L^S$ be a set of S-formulas. If for every $n \in \mathbb{N}$ there is an S-interpretation with domain of size at least n which models Φ, then there is an S-interpretation with infinite domain which models Φ.*

Proof. For $n \in \mathbb{N}$, define

$$\phi_n := \exists v_0 \cdots \exists v_{n-1} (\neg v_0 \equiv v_1 \wedge \cdots \wedge \neg v_0 \equiv v_{n-1} \cdots \wedge \neg v_{n-1} \equiv v_0 \wedge \cdots \wedge v_{n-1} \equiv v_{n-1}),$$

and let

$$\Psi = \Phi \cup \{\phi_n;\, n \geq 12\}.$$

We claim $\operatorname{Sat}\Psi$. Let $\Psi_0 \subseteq \Psi$ be finite. Then $\Psi_0 \subseteq \Phi \cup \{\phi_{12}, \ldots, \phi_N\}$ for some $N \in \mathbb{N}$. We are given that there exists a model $\mathfrak{I}$ of Φ with domain which has at least N elements. Because the domain has size at least N, we get $\mathfrak{I} \models \phi_n$ for all $n \leq N$. Therefore $\mathfrak{I} \models \Psi_0$ and thus $\operatorname{Sat}\Psi_0$. By the compactness theorem, we have $\operatorname{Sat}\Psi$.

Hence there is a model $\mathfrak{I}$ of Ψ. For $n \in \mathbb{N}$, since $\mathfrak{I} \models \phi_n$ we know that the domain of $\mathfrak{I}$ has size at least n. As $n \in \mathbb{N}$ was arbitrary, the domain must be infinite. Finally, note $\mathfrak{I} \models \Phi$ because $\Phi \subseteq \Psi$. ∎

THEOREM 89. [UPWARD LÖWENHEIM-SKOLEM THEOREM] *Let S be a first-order language, let $\Phi \subseteq L^S$ be a set of S-formulas and let A be a set. If Φ is satisfiable over an infinite domain then there is a model of Φ whose domain has cardinality at least $|A|$.*

Proof. For $a \in A$, let c_a^S be a distinct constant symbol not already in S. Let

$$\Psi = \Phi \cup \{\neg c_a \equiv c_b;\, a, b \in A,\ a \neq b\}.$$

We claim that Ψ is satisfiable. By the compactness theorem, it suffices to show that any finite subset is satisfiable. So let $\Psi_0 \subseteq \Psi$ be finite. Then $\Psi_0 \subseteq \Phi \cup \{\neg c_{a_i} \equiv c_{a_j};\, i, j \in \{1, \ldots, n\},\ i \neq j\}$ for some distinct $a_1, \ldots, a_n \in A$. We are given an S-interpretation $((B, \mathfrak{b}), \beta)$ which satisfies Φ and such that B is infinite. As such, we can select distinct $b_1, \ldots, b_n \in B$. Extend $\mathfrak{b}$ to $\mathfrak{a}$ defined on $S \cup \{c_{a_1}, \ldots, c_{a_n}\}$ by $\mathfrak{a}(c_{a_i}) = b_i$ for $i \in \{1, \ldots, n\}$. By the coincidence lemma, $((B, \mathfrak{a}), \beta) \models \Phi$. Moreover, $((B, \mathfrak{a}), \beta) \models \neg c_{a_i} \equiv c_{a_j}$ for $i, j \in \{1, \ldots, n\}$ for which $i \neq j$. Therefore $((B, \mathfrak{a}), \beta) \models \Psi_0$ and thus $\operatorname{Sat}\Psi_0$.

This means that there is an $S \cup \{c_a;\, a \in A\}$-interpretation $\mathfrak{I}$ which satisfies Ψ. As $\Phi \subseteq \Psi$, $\mathfrak{I} \models \Phi$. We need only find an injection from A to the domain of $\mathfrak{I}$. Write $\mathfrak{I} = ((C, \mathfrak{c}), \gamma)$. Define $\pi : A \to C$ by $\pi(a) = \mathfrak{c}(c_a)$. This map is injective because it models the sentences we added to Φ to create Ψ. ∎

6.3 Elementary Classes

DEFINITION 90. Let S be a first-order language and let $\Phi \subseteq L_0^S$ be a set of S-sentences. We define the *class of models of* Φ to be

$$\operatorname{Mod}^S \Phi := \{\mathfrak{A};\, \mathfrak{A} \text{ is an S-structure with } \mathfrak{A} \models \Phi\}.$$

REMARK **91.** Note $\mathrm{Mod}^S \Phi$ is not a set; it is a class (whatever that is..).

DEFINITION **92.** Let S be a first-order language and let $\mathfrak{K}$ be a class of S-structures. We say $\mathfrak{K}$ is *elementary* if there is an S-sentence $\phi \in L_0^S$ such that $\mathfrak{K} = \mathrm{Mod}^S\{\phi\}$. We say $\mathfrak{K}$ is Δ-*elementary* if there is a set of S-sentences $\Phi \subseteq L_0^S$ such that $\mathfrak{K} = \mathrm{Mod}^S \Phi$.

REMARK **93.** Clearly any elementary class is a Δ-elementary class. On the other hand, any Δ-elementary class is an intersection of elementary classes. Indeed,

$$\mathrm{Mod}^S \Phi = \bigcap_{\phi \in \Phi} \mathrm{Mod}^S\{\phi\}.$$

DEFINITION **94.** Let $S_{\mathrm{ar}} := \{+, \cdot, 0, 1\}$ be the language of arithmetic. Additionally, set $S_{\mathrm{ar}}^< := \{+, \cdot, 0, 1, <\}$.

Define $\mathfrak{n}(+)$ to be usual addition on $\mathbb{N}$, $\mathfrak{n}(\cdot)$ to be usual multiplication on $\mathbb{N}$, $\mathfrak{n}(0) := \varnothing$ (the number zero), and $\mathfrak{n}(1) := \{\varnothing\}$ (the number one). We set

$$\mathfrak{N} := (\mathbb{N}, \mathfrak{n}),$$

an S_{ar}-structure. Extend $\mathfrak{n}$ to $\mathfrak{n}^<$ with $\mathfrak{n}^<(<)$ being the usual order on $\mathbb{N}$. We set

$$\mathfrak{N}^< := (\mathbb{N}, \mathfrak{n}^<),$$

an $S_{\mathrm{ar}}^<$-structure.

EXAMPLE **95.** Let

$$\begin{aligned}
\phi_1 &:= \forall v_0 \forall v_1 \forall v_2 + +v_0 v_1 v_2 \equiv +v_0 + v_1 v_2 \\
\phi_2 &:= \forall v_0 + v_0 0 \equiv v_0 \\
\phi_3 &:= \forall v_0 \forall v_1 \forall v_2 \cdot \cdot v_0 v_1 v_2 \equiv \cdot v_0 \cdot v_1 v_2 \\
\phi_4 &:= \forall v_0 \cdot v_0 1 \equiv v_0 \\
\phi_5 &:= \forall v_0 \exists v_1 + v_0 v_1 \equiv 0 \\
\phi_6 &:= \forall v_0 (\neg v_0 \equiv 0 \to \exists v_1 \cdot v_0 v_1 \equiv 1) \\
\phi_7 &:= \forall v_0 \forall v_1 + v_0 v_1 \equiv +v_1 v_0 \\
\phi_8 &:= \forall v_0 \forall v_1 \cdot v_0 v_1 \equiv \cdot v_1 v_0 \\
\phi_9 &:= \neg 0 \equiv 1 \\
\phi_{10} &:= \forall v_0 \forall v_1 \forall v_2 \cdot v_0 + v_1 v_2 \equiv + \cdot v_0 v_1 \cdot v_0 v_2
\end{aligned}$$

be the field axioms, and let

$$\phi_F := \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6 \wedge \phi_7 \wedge \phi_8 \wedge \phi_9 \wedge \phi_{10}.$$

Then $\mathrm{Mod}^{S_{ar}}\{\phi_F\}$ is the class of fields.

Now for $p \in \mathbb{N}$ prime, let

$$\chi_p := +\cdots+11\cdots1 \equiv 0,$$

with p 1's and $p-1$ +'s. Using our abusive notation for 2-ary relation symbols, this can be written as $1+\cdots+1 \equiv 0$ with p 1's.

Recall that a field $\mathfrak{F}$ has characteristic p, for a prime p, if $\mathfrak{F} \vDash \chi_p$, and that it has characteristic zero if $\mathfrak{F} \vDash \neg\chi_q$ for all primes $q \in \mathbb{N}$. Therefore

$$\mathfrak{F}_p := \mathrm{Mod}^{S_{ar}}\{\phi_F \wedge \chi_p\}$$

is the class of fields of characteristic p, and is hence elementary. Moreover

$$\mathfrak{F}_0 := \mathrm{Mod}^{S_{ar}}\{\phi_F\} \cup \{\neg\chi_p; p \in \mathbb{N} \text{ prime}\}$$

is the class of fields of characteristic zero, and is hence Δ-elementary.

Proposition 96. *The class $\mathfrak{F}_0$ of fields of characteristic zero is not elementary.*

Proof. Assume, for a contradiction, that there exists an S_{ar}-sentence ϕ such that $\mathfrak{F}_0 = \mathrm{Mod}^{S_{ar}}\{\phi\}$. Then $\{\phi_F\} \cup \{\neg\chi_p; p \in \mathbb{N} \text{ prime}\} \vDash \phi$. By compactness, there is some prime $q \in \mathbb{N}$ such that $\{\phi_F\} \cup \{\neg\chi_p; p < q \text{ prime}\} \vDash \phi$. Consider the field $\mathbb{Z}/q\mathbb{Z}$. As it is a field, we have $\mathbb{Z}/q\mathbb{Z} \vDash \phi_F$. Moreover, $\mathbb{Z}/q\mathbb{Z} \vDash \neg\chi_p$ for all $p < q$. Therefore $\mathbb{Z}/q\mathbb{Z} \vDash \phi$. Hence $\mathbb{Z}/q\mathbb{Z} \in \mathfrak{F}_0$ is a field of characteristic zero. This is absurd. ∎

6.4 Elementarily Equivalent Structures

Definition 97. Let S be a first-order language and let $\mathfrak{A}$ and $\mathfrak{B}$ be S-structures. We say $\mathfrak{A}$ and $\mathfrak{B}$ are *elementarily equivalent*, and write $\mathfrak{A} \equiv \mathfrak{B}$, if for every S-sentence $\phi \in L_0^S$ we have $\mathfrak{A} \vDash \phi$ if and only if $\mathfrak{B} \vDash \phi$.

Definition 98. Let S be a first-order language and let $\mathfrak{A}$ be an S-structure. The *theory of* $\mathfrak{A}$ is

$$\mathrm{Th}(\mathfrak{A}) := \{\phi \in L_0^S; \mathfrak{A} \vDash \phi\}.$$

Remark 99. The following is just a restatement of the isomorphism lemma in terms of our new terminology.

Lemma 100. *Let S be a first-order language and let $\mathfrak{A}$ be an S-structure. Then*

$$\{\mathfrak{B}; \mathfrak{B} \cong \mathfrak{A}\} \subseteq \{\mathfrak{B}; \mathfrak{B} \equiv \mathfrak{A}\}.$$

Proof. Suppose $\mathfrak{B} \cong \mathfrak{A}$. Let $\phi \in L_0^S$ be an S-sentence. By the isomorphism lemma, we have $\mathfrak{A} \vDash \phi$ if and only if $\mathfrak{B} \vDash \phi$. Thus $\mathfrak{B} \equiv \mathfrak{A}$. ∎

LEMMA **101.** *Let S be a first-order language and let $\mathfrak{A}$ be an S-structure. Then*

$$\operatorname{Mod}^S \operatorname{Th}(\mathfrak{A}) = \{\mathfrak{B};\, \mathfrak{B} \equiv \mathfrak{A}\}.$$

Equivalently, for an S-structure $\mathfrak{B}$, we have $\mathfrak{B} \equiv \mathfrak{A}$ if and only if $\mathfrak{B} \models \operatorname{Th}(\mathfrak{A})$.

Proof. Let $\mathfrak{B}$ be an S-structure and suppose $\mathfrak{B} \equiv \mathfrak{A}$. Suppose $\phi \in \operatorname{Th}(\mathfrak{A})$. Then $\mathfrak{A} \models \phi$. Since they are elementarily equivalent, we get $\mathfrak{B} \models \phi$. Therefore $\mathfrak{B} \models \operatorname{Th}(\mathfrak{A})$, and thus $\mathfrak{B} \in \operatorname{Mod}^S \operatorname{Th}(\mathfrak{A})$.

Conversely, let $\mathfrak{B}$ be an S-structure and suppose $\mathfrak{B} \in \operatorname{Mod}^S \operatorname{Th}(\mathfrak{A})$. This means $\mathfrak{B} \models \operatorname{Th}(\mathfrak{A})$. Now let $\phi \in L_0^S$ be an S-sentence. If $\mathfrak{A} \models \phi$ then $\phi \in \operatorname{Th}(\mathfrak{A})$, and hence $\mathfrak{B} \models \phi$. Now suppose $\mathfrak{B} \models \phi$ and assume, for a contradiction, we do not have $\mathfrak{A} \models \phi$. Then $\mathfrak{A} \models \neg\phi$ and hence $\neg\phi \in \operatorname{Th}(\mathfrak{A})$. Therefore $\mathfrak{B} \models \neg\phi$ and we conclude we do not have $\mathfrak{B} \models \phi$. Contradiction. Hence $\mathfrak{A} \models \phi$. Thus $\mathfrak{A} \models \phi$ if and only if $\mathfrak{B} \models \phi$, and so $\mathfrak{A} \equiv \mathfrak{B}$. ∎

THEOREM **102.** *Let S be a first-order language and let $\mathfrak{A}$ be an S-structure.*

(a) If the domain of $\mathfrak{A}$ is infinite then the class $\{\mathfrak{B};\, \mathfrak{B} \cong \mathfrak{A}\}$ of S structures which are isomorphic to $\mathfrak{A}$ is not Δ-elementary.

(b) The class $\{\mathfrak{B};\, \mathfrak{B} \equiv \mathfrak{A}\}$ of S-structures which are elementarily equivalent to $\mathfrak{A}$ is Δ-elementary, and moreover, this is the smallest Δ-elementary class which contains $\mathfrak{A}$.

Proof. (a) Suppose the domain of $\mathfrak{A}$ is infinite. Assume, for a contradiction, that there is some set $\Phi \subseteq L_0^S$ of S-sentences such that $\{\mathfrak{B};\, \mathfrak{B} \cong \mathfrak{A}\} = \operatorname{Mod}^S \Phi$. Note Φ is satisfiable by $\mathfrak{A}$, which has infinite domain. By the upward Löwenheim-Skolem theorem, there is a model $\mathfrak{B}$ of Φ whose domain has cardinality $\mathcal{P}(A)$, where A is the domain of $\mathfrak{A}$. We cannot have $\mathfrak{B} \cong \mathfrak{A}$ because this would imply that their domains have the same cardinality, which they don't by construction. Hence $\mathfrak{B} \in \operatorname{Mod}^S \Phi$ yet $\mathfrak{B} \ncong \mathfrak{A}$. Contradiction. Hence $\{\mathfrak{B};\, \mathfrak{B} \cong \mathfrak{A}\}$ is not Δ-elementary.

(b) By the previous lemma we have $\{\mathfrak{B};\, \mathfrak{B} \equiv \mathfrak{A}\} = \operatorname{Mod}^S \operatorname{Th}(\mathfrak{A})$, and hence it is Δ-elementary.

Now suppose we have a Δ-elementary class of S-structures, $\mathfrak{K}$, with $\mathfrak{A} \in \mathfrak{K}$. Then there exists a set of S-sentences $\Phi \subseteq L_0^S$ such that $\mathfrak{K} = \operatorname{Mod}^S \Phi$. Suppose $\mathfrak{B}$ is an S-structure with $\mathfrak{B} \equiv \mathfrak{A}$. Let $\phi \in \Phi$. Then $\mathfrak{A} \models \phi$, and therefore $\mathfrak{B} \models \phi$. This means $\mathfrak{B} \models \Phi$ and hence $\mathfrak{B} \in \operatorname{Mod}^S \Phi = \mathfrak{K}$. Thus $\{\mathfrak{B};\, \mathfrak{B} \equiv \mathfrak{A}\} \subseteq \mathfrak{K}$.

∎

DEFINITION **103.** An S_{ar}-structure which is elementarily equivalent to $\mathfrak{N}$, but not isomorphic to $\mathfrak{N}$, is called a *non-standard model of arithmetic.*

Theorem 104. [Skolem's Theorem] *There exists a non-standard model of arithmetic whose domain is countable.*

Proof. Define

$$\Psi = \mathrm{Th}(\mathfrak{N}) \cup \{\neg v_0 \equiv \underline{n};\, n \in \mathbb{N}\},$$

where, for $n \in \mathbb{N}$ and $n \neq 0$, $\underline{n} = +\cdots+11\cdots1$ (with $n-1$ +'s and n 1's), and $\underline{0} = 0$. We claim $\mathrm{Sat}\,\Psi$. By the compactness theorem it suffices to show $\mathrm{Sat}\,\Psi_0$ for all finite subsets $\Psi_0 \subseteq \Psi$. To this end, let $\Psi_0 \subseteq \Psi$ be finite. Then $\Psi \subseteq \mathrm{Th}(\mathfrak{N}) \cup \{\neg v_0 \equiv 0, \ldots, \neg v_0 \equiv \underline{N}\}$ for some $N \in \mathbb{N}$. Let β be any assignment in $\mathbb{N}$. Then $(\mathfrak{N}, \beta\frac{N+1}{v_0}) \models \mathrm{Th}(\mathfrak{N})$ already, and moreover we see $(\mathfrak{N}, \beta\frac{N+1}{v_0}) \models \neg v_0 \equiv \underline{n}$ for all $n \leq N$. Therefore $(\mathfrak{N}, \beta\frac{N+1}{v_0}) \models \Psi_0$ and so $\mathrm{Sat}\,\Psi_0$. Thus $\mathrm{Sat}\,\Psi$.

By the Löwenheim-Skolem theorem, there is a model $(\mathfrak{A}, \beta')$ of Ψ whose domain is countable (indeed, Ψ is countable because S_{ar} is countable). In particular, $\mathfrak{A} \models \mathrm{Th}(\mathfrak{N})$. As we just saw, this means $\mathfrak{A} \equiv \mathfrak{N}$. We need only show $\mathfrak{A} \ncong \mathfrak{N}$.

Assume, for a contradiction, there exists an isomorphism $\pi : \mathfrak{N} \to \mathfrak{A}$. By the definition of an isomorphism, we find $\pi(0) = 0^{\mathfrak{A}}$ and $\pi(1) = 1^{\mathfrak{A}}$. Moreover, for all $n \in \mathbb{N}$, we have $\pi(n) = \underline{n}^{\mathfrak{A}}$ (check this). But $\beta'(v_0) \in A$ with $\beta'(v_0) \neq \underline{n}^{\mathfrak{A}}$ for all $n \in \mathbb{N}$. Thus π is not surjective. Contradiction, as required. ∎

Theorem 105. *There is a non-standard model of* $\mathrm{Th}(\mathfrak{N}^<)$ *whose domain is countable.*

Proof. Exercise — similar to above. ∎

Remark 106. One can explore what such a non-standard model of $\mathrm{Th}(\mathfrak{N}^<)$ "looks like". See the book.

7 The Scope of First-Order Logic

8 Syntactic Interpretations and Normal Forms

8.1 Term-Reduced Formulas and Relational Symbol Sets

Definition 107. Let S be a first-order language and let $\phi \in L^S$ be an S-formula. We say ϕ is *term-reduced* if all of its atomic subformulas are among the following forms:

1. $Rx_1\cdots x_n$ for $n \in \mathbb{N}$, n-ary relation symbol $R \in S$, and variables $x_1, \ldots, x_n$;
2. $x \equiv y$ for variables x and y;
3. $fx_1\cdots x_n \equiv x$ for $n \in \mathbb{N}$, n-ary function symbol $f \in S$, and variables $x_1, \ldots, x_n, s$;
4. $c \equiv x$ for constant symbol $c \in S$ and variable x.

THEOREM 108. *Let S be a first-order language and let $\psi \in L^S$ be an S-formula. Then there is a term-reduced S-formula $[\psi]^* \in L^S$ such that $[\psi]^* \dashv\vDash \psi$ and* $\text{free}_S(\psi) = \text{free}_S([\psi]^*)$.

Proof. We define $[\psi]^*$ recursively. For $\psi \in L^S$, let $x_1^\psi, x_2^\psi, \ldots$ list in order the variables not in $\text{var}_S(\psi)$. Define:

$[y \equiv x]^* := y \equiv x$ for variables x and y;

$[c \equiv x]^* := c \equiv x$ for constant symbol $c \in S$ and variable x;

$[ft_1 \cdots t_n \equiv x]^* := \exists x_1^{ft_1\cdots t_n \equiv x} \cdots \exists x_n^{ft_1\cdots t_n \equiv x}([t_1 \equiv x_1^{ft_1\cdots t_n \equiv x}]^* \wedge \cdots \wedge [t_n \equiv x_n^{ft_1\cdots t_n \equiv x}]^* \wedge f x_1^{ft_1\cdots t_n \equiv x} \cdots x_n^{ft_1\cdots t_n \equiv x} \equiv x)$ for $n \in \mathbb{N}$, S-terms $t_1, \ldots, t_n \in T^S$, n-ary function symbol $f \in S$, and variable x;

$[t_1 \equiv t_2]^* := \exists x_1^{t_1 \equiv t_2}([t_1 \equiv x_1^{t_1 \equiv t_2}]^* \wedge [t_2 \equiv x_1^{t_1 \equiv t_2}]^*)$ for S-terms t_1 and t_2, where t_2 is not a variable;

$[Rt_1 \cdots t_n]^* := \exists x_1^{Rt_1\cdots t_n} \cdots \exists x_n^{Rt_1\cdots t_n}([t_1 \equiv x_1^{Rt_1\cdots t_n}]^* \wedge \cdots \wedge [t_n \equiv x_n^{Rt_1\cdots t_n}]^* \wedge R x_1^{Rt_1\cdots t_n} \cdots x_n^{Rt_1\cdots t_n})$ for $n \in \mathbb{N}$, S-terms $t_1, \ldots, t_n \in T^S$, and n-ary relation symbol $R \in S$;

$[\neg\phi]^* := \neg[\phi]^*$ for S-formula $\phi \in L^S$;

$[(\psi_1 \vee \psi_2)]^* := ([\psi_1]^* \vee [\psi_2]^*)$ for S-formulas $\psi_1, \psi_2 \in L^S$;

$[\exists x\phi]^* := \exists x[\phi]^*$ for S-formula $\phi \in L^S$ and variable x.

The desired result now follows from a simple induction — exercise. ∎

DEFINITION 109. Let S be a first-order language. We say S is *relational* if it contains only relation symbols.

DEFINITION 110. Let S be a first-order language. Define a symbol set S^r to consist of the relation symbols in S, along with:

for every $n \in \mathbb{N}$ and n-ary relation symbol $f \in S$, an $n+1$-ary relation symbol F_f;

and for every constant symbol $c \in S$, a 1-ary relation symbol C_c.

REMARK 111. If S is a first-order language then S^r is a first-order language which is relational.

DEFINITION 112. Let S be a first-order language and let $(A, \mathfrak{a})$ be an S-structure. Define $\mathfrak{a}^r$ on S^r by:

$\mathfrak{a}^r(R) := \mathfrak{a}(R)$ for relation symbols $R \in S$;

$a^r(F_f) := \{(a_1, \ldots, a_n, a(f)((a_1, \ldots, a_n))); (a_1, \ldots, a_n) \in A^n\}$ for $n \in \mathbb{N}$ and n-ary function symbols $f \in S$;

$a^r(C_c) := \{(a(c))\}$ for constant symbol $c \in S$.

We let $(A, a)^r := (A, a^r)$.

REMARK **113.** If S is a first-order language and $\mathfrak{A}$ is an S-structure then $\mathfrak{A}^r$ is an S^r-structure.

THEOREM **114.** *Let S be a first-order language.*

(a) For every S-formula $\psi \in L_n^S$ there is an S^r-formula $[\psi]^r \in L_n^{S^r}$ such that for all S-interpretations $(\mathfrak{A}, \beta)$ we have $(\mathfrak{A}, \beta) \models \psi$ if and only if $(\mathfrak{A}^r, \beta) \models [\psi]^r$.

(b) For every S^r-formula $\psi \in L_n^{S^r}$ there is an S-formula $[\psi]^{-r} \in L_n^S$ such that for all S-interpretations $(\mathfrak{A}, \beta)$ we have $(\mathfrak{A}, \beta) \models [\psi]^{-r}$ if and only if $(\mathfrak{A}^r, \beta) \models \psi$.

Proof. (a) Let $\psi \in L^S$ be an S-formula. By the previous theorem, there is a formula $\phi \in L^S$ which is term-reduced and is logically equivalent to ψ, with $\text{free}_S(\psi) = \text{free}_S(\phi)$. In particular, for any S-interpretation $(\mathfrak{A}, \beta)$, we have $(\mathfrak{A}, \beta) \models \psi$ if and only if $(\mathfrak{A}, \beta) \models \phi$. Assuming we have defined and proved the result for term-reduced formulas, we can let $[\psi]^r := [\phi]^r$. By assumption, for all S-interpretations $(\mathfrak{A}, \beta)$ we have $(\mathfrak{A}, \beta) \models \phi$ if and only if $(\mathfrak{A}^r, \beta) \models [\phi]^r$. Combining this with $\psi \mathrel{=\!\!|\!\models} \phi$, we get the desired result.

Therefore it suffices to prove the result for term-reduced formulas. So suppose $\psi \in L^S$ is a term-reduced S-formula. We proceed by induction on term-reduced formulas to define $[\psi]^r$:

$[Rx_1 \cdots x_n]^r := Rx_1 \cdots x_n$ for $n \in \mathbb{N}$, n-ary relation symbol $R \in S$, and variables $x_1, \ldots, x_n$;

$[x \equiv y]^r := x \equiv y$ for variables x and y;

$[fx_1 \cdots x_n \equiv x]^r := F_f x_1 \cdots x_n x$ for $n \in \mathbb{N}$, n-ary function symbol $f \in S$, and variables $x_1, \ldots, x_n, x$;

$[c \equiv x]^r := C_c x$ for constant symbol $c \in S$ and variable x;

$[\neg\chi]^r := \neg[\chi]^r$ for term-reduced $\chi \in L^S$;

$[(\chi_1 \vee \chi_2)]^r := ([\chi_1]^r \vee [\chi_2]^r)$ for term-reduced $\chi_1, \chi_2 \in L^S$;

$[\exists x\chi]^r := \exists x[\chi]^r$ for term-reduced $\chi \in L^S$ and variable x.

Indeed, with this definition in place, it is easy to show, via induction, that for all S-interpretations $(\mathfrak{A}, \beta)$ we have $(\mathfrak{A}, \beta) \models \psi$ if and only if $(\mathfrak{A}^r, \beta) \models [\psi]^r$ — exercise.

(b) This is similar to part (a), we only note that in particular we will define

$$[F_f t_1 \cdots t_n t]^{-r} := f t_1 \cdots t_n \equiv t;$$
$$[C_c t]^{-r} := c \equiv t.$$

The details are left as an exercise.

■

COROLLARY 115. *Let S be a first-order language and let $\mathfrak{A}$ and $\mathfrak{B}$ be S-structures. Then $\mathfrak{A} \equiv \mathfrak{B}$ if and only if $\mathfrak{A}^r \equiv \mathfrak{B}^r$.*

Proof. Suppose first that $\mathfrak{A} \equiv \mathfrak{B}$ and let $\psi \in L_0^{S^r}$ be an S^r-sentence. By the theorem, there exists an S-sentence $[\psi]^{-r} \in L_0^S$ such that for all S-interpretations $(\mathfrak{A}, \beta)$ we have $(\mathfrak{A}, \beta) \models [\psi]^{-r}$ if and only if $(\mathfrak{A}^r, \beta) \models \psi$. So we see $\mathfrak{A}^r \models \psi$ if and only if $\mathfrak{A} \models [\psi]^{-r}$ if and only if $\mathfrak{B} \models [\psi]^{-r}$ if and only if $\mathfrak{B}^r \models \psi$. Therefore $\mathfrak{A}^r \equiv \mathfrak{B}^r$.

Conversely, suppose $\mathfrak{A}^r \equiv \mathfrak{B}^r$ and let $\psi \in L_0^S$ be an S-sentence. By the theorem, there exists an S^r-sentence $[\psi]^r \in L_0^{S^r}$ such that for all S-interpretations $(\mathfrak{A}, \beta)$ we have $(\mathfrak{A}, \beta) \models \psi$ if and only if $(\mathfrak{A}^r, \beta) \models [\psi]^r$. So we see $\mathfrak{A} \models \psi$ if and only if $\mathfrak{A}^r \models [\psi]^r$ if and only if $\mathfrak{B}^r \models [\psi]^r$ if and only if $\mathfrak{B} \models \psi$. Therefore $\mathfrak{A} \equiv \mathfrak{B}$. ■

8.2 Syntactic Interpretations

8.3 Extensions by Definition

8.4 Normal Forms

DEFINITION 116. Let S be a first-order language and let $\phi \in L^S$ be an S-formula. We say ϕ is in disjunctive normal form if there exist $\phi_{0,0}, \ldots, \phi_{0,b}, \ldots, \phi_{a,0}, \ldots, \phi_{a,b} \in L^S$ such that each $\phi_{i,j}$ is either atomic or the negation of an atomic formula, and such that

$$\phi = ((\phi_{0,0} \wedge \cdots \wedge \phi_{0,b}) \vee \cdots \vee (\phi_{a,0} \wedge \cdots \wedge \phi_{a,b})).$$

DEFINITION 117. Let S be a first-order language and let $\phi \in L^S$ be an S-formula. We define $qn_S : L^S \to \mathbb{N}$ by

$qn_S(t_1 \equiv t_2) := 0$ for S-terms t_1 and t_2;

$qn_S(Rt_1 \cdots t_n) := 0$ for $n \in \mathbb{N}$, n-ary relations symbol $R \in S$, and S-terms $t_1, \ldots, t_n \in T^S$;

$qn_S(\neg\psi) := qn_S(\psi)$ for S-formula $\psi \in L^S$;

$qn_S((\psi_1 \vee \psi_2)) := qn_S(\psi_1) + qn_S(\psi_2)$ for S-formulas $\psi_1, \psi_2 \in L^S$;

$qn_S(\exists x \psi) := qn_S(\psi) + 1$ for S-formula $\psi \in L^S$ and variable x.

We say ϕ is *quantifier-free* if $\mathrm{qn}_S(\phi) = 0$.

THEOREM 118. *Let S be a first-order language and let $\phi \in L^S$ be an S-formula. If ϕ is quantifier-free then ϕ is logically equivalent to a formula in disjunctive normal form.*

Proof. Let $\phi \in L^S$ be an S-formula. Then there exists some $k \in \mathbb{N}$ such that $\phi \in L^S_k$. Let $\phi_0, \dots, \phi_n$ be the atomic formulas which appear in ϕ. For an S-interpretation $\mathfrak{A} = (A, \mathfrak{a})$ and $(a_0, \dots, a_{k-1}) \in A^k$, set

$$\psi_{(\mathfrak{A},(a_0,\dots,a_{k-1}))} := \psi_0 \wedge \cdots \wedge \psi_n,$$

where

$$\psi_i := \begin{cases} \phi_i & \text{if } \mathfrak{A} \vDash \phi_i[a_0, \dots, a_{k-1}] \\ \neg\phi_i & \text{if } \mathfrak{A} \vDash \neg\phi_i[a_0, \dots, a_{k-1}]. \end{cases}$$

Note that $\mathfrak{A} \vDash \psi_{(\mathfrak{A},(a_0,\dots,a_{k-1}))}[a_0, \dots, a_{k-1}]$ for S-interpretation $\mathfrak{A} = (A, \mathfrak{a})$ and $(a_0, \dots, a_{k-1}) \in A^k$. Moreover,

$$\left|\{\psi_{(\mathfrak{A},(a_0,\dots,a_{k-1}))};\, \mathfrak{A} = (A, \mathfrak{a}) \text{ is an S-structure and } (a_0, \dots, a_{k-1}) \in A^k \text{ with } \mathfrak{A} \vDash \phi[a_0, \dots, a_{k-1}]\}\right| \leq 2^{n+1},$$

and so we can list it as $\chi_1, \dots, \chi_m$. Let

$$\chi := \chi_1 \vee \cdots \vee \chi_m.$$

Note that χ is in disjunctive normal form.

We claim $\phi \dashv\vDash \chi$. Suppose $\mathfrak{B} = (B, \mathfrak{b})$ is an S-interpretation and $(b_0, \dots, b_{k-1}) \in B^k$ is such that $\mathfrak{B} \vDash \phi[b_0, \dots, b_{k-1}]$. Then $\psi_{(\mathfrak{B},(b_0,\dots,b_{k-1})} = \chi_i$ for some $i \in \{1, \dots, m\}$. By construction, we have $\mathfrak{B} \vDash \phi[b_0, \dots, b_{k-1}]$, and hence $\mathfrak{B} \vDash \chi_i[b_0, \dots, b_{k-1}]$. By definition we therefore have $\mathfrak{B} \vDash \chi[b_0, \dots, b_{k-1}]$.

Conversely, suppose $\mathfrak{B} = (B, \mathfrak{b})$ is an S-interpretation and $(b_0, \dots, b_{k-1}) \in B^k$ is such that $\mathfrak{B} \vDash \chi[b_0, \dots, b_{k-1}]$. Then $\mathfrak{B} \vDash \chi_i[b_0, \dots, b_{k-1}]$ for some $i \in \{1, \dots, m\}$. Write $\chi_i = \psi_{(\mathfrak{A},(a_0,\dots,a_{k-1}))}$ for some S-structure $\mathfrak{A}$ and some $(a_0, \dots, a_{k-1}) \in A^k$. So we have $\mathfrak{B} \vDash \psi_{(\mathfrak{A},(a_0,\dots,a_{k-1}))}[b_0, \dots, b_{k-1}]$. By definition of $\psi_{(\mathfrak{A},(a_0,\dots,a_{k-1}))}$, we can see that this means $\mathfrak{B} \vDash \phi_j[b_0, \dots, b_{k-1}]$ if and only if $\mathfrak{A} \vDash \phi_j[a_0, \dots, a_{k-1}]$, for each $j \in \{0, \dots, n\}$. Because ϕ is obtained from $\phi_0, \dots, \phi_n$ via only disjunctions and negations (since $\mathrm{qn}_S(\phi) = 0$), it follows from a simple induction that $\mathfrak{A} \vDash \phi[a_0, \dots, a_{k-1}]$ if and only if $\mathfrak{B} \vDash \phi[b_0, \dots, b_{k-1}]$ (check). But we know $\mathfrak{A} \vDash \phi[a_0, \dots, a_{k-1}]$, and thus we may conclude $\mathfrak{B} \vDash \phi[b_0, \dots, b_{k-1}]$.

Therefore $\phi \dashv\vDash \chi$ and χ is in disjunctive normal form. ∎

DEFINITION 119. Let S be a first-order language and let $\phi \in L^S$ be an S-formula. We say ϕ is in conjunctive normal form if there exist $\phi_{0,0}, \dots, \phi_{0,b}, \dots, \phi_{a,0}, \dots, \phi_{a,b} \in L^S$ such that each $\phi_{i,j}$ is either atomic or the negation of an atomic formula, and such that

$$\phi = ((\phi_{0,0} \vee \cdots \vee \phi_{0,b}) \wedge \cdots \wedge (\phi_{a,0} \vee \cdots \vee \phi_{a,b})).$$

DEFINITION **120.** Let S be a first-order language and let $\phi \in L^S$ be an S-formula. We say ϕ is in *prenex normal form* if it is of the form

$$\phi = Q_0 x_0 \cdots Q_m x_m \psi$$

for some $Q_0, \ldots, Q_m \in \{\forall, \exists\}$, some variables $x_0, \ldots, x_m$, and some quantifier-free S-formula $\psi \in L^S$.

THEOREM **121.** *Let S be a first-order language and let $\phi \in L^S$ be an S-formula. Then ϕ is logically equivalent to an S-formula $\psi \in L^S$ in prenex normal form with* $\text{free}_S(\phi) = \text{free}_S(\psi)$.

Proof. Exercise — see class notes. ∎

9 Extensions of First-Order Logic

10 Limitations of the Formal Method

10.1 Decidability and Enumerability

REMARK **122.** We will work at a relatively informal level when it comes to procedures. The essential ideas of a procedure are as follows.

A procedure may run on inputs of words over a language. It may output. It may halt. Basically, it is a finite set of instructions.

We will be more precise when we get to a particular type of procedure, namely register machines.

REMARK **123.** If we have a procedure $\mathfrak{P}$, then since it is essentially a finite set of instructions, it can only refer to finitely many symbols. This is not very compatible with infinite alphabets. If we have a countable alphabet, then this can easily be encoded in a new finite alphabet by putting $\overline{0}, \ldots, \overline{9}, \underline{0}, \ldots, \underline{9}$ into the new language and interpreting, for example, $v\underline{1}\,\underline{2}$ as v_{12}.

As such, we will only consider finite alphabets in the rest of this section, while keeping in mind that what we do can be applied to countable alphabets.

DEFINITION **124.** Let $\mathcal{A}$ be an alphabet, let $W \subseteq \mathcal{A}^*$ be a set of words over $\mathcal{A}$, and let $\mathfrak{P}$ be a procedure. We say $\mathfrak{P}$ is a *decision procedure* for W if for every input $x \in \mathcal{A}^*$, $\mathfrak{P}$ eventually stops, having (before stopping) output exactly one word $y \in \mathcal{A}^*$, where $y = \square$ if $x \in W$ and $y \neq \square$ if $x \notin W$.

We say W is *decidable* if there is a decision procedure for W.

Definition 125. Let $\mathcal{A}$ be a finite alphabet, let $W \subseteq \mathcal{A}^*$ be a set of words over $\mathcal{A}$, and let $\mathfrak{P}$ be a procedure. We say $\mathfrak{P}$ is an *enumeration procedure* for W if $\mathfrak{P}$, once having been started, eventually outputs exactly the words in W (possibly with repetition).

We say W is *enumerable* if there is an enumeration procedure for W.

Definition 126. Let $\mathcal{A} = \{a_0, \dots, a_n\}$ be a finite alphabet (with a given order) and let $x, y \in \mathcal{A}^*$ be words over $\mathcal{A}$. We write $x <_L y$ if either $\mathrm{len}(x) < \mathrm{len}(y)$, or $\mathrm{len}(x) = \mathrm{len}(y)$ and there exist $i, j \in \{0, \dots, n\}$ and $z, w, w' \in \mathcal{A}^*$ such that $i < j$ and $x = za_iw$ and $y = za_jw'$. Then $<_L$ is called the *lexicographical ordering* on $\mathcal{A}^*$.

Example 127. If $\mathcal{A} = \{x, y, z\}$ then $\square <_L z$, $z <_L xy$, and $xyzyxy <_L xyzyyx$.

Example 128. Let $\mathcal{A}$ be a finite alphabet. Then $\mathcal{A}^*$ is enumerable. Indeed, there is clearly a procedure for listing out the elements in $\mathcal{A}^*$ in lexicographical order.

Example 129. The set $\{\phi \in L_0^{S_\infty}; \models \phi\} \subseteq \mathcal{A}_\infty$ is enumerable. Indeed, note that this is just the set $\{\phi \in L_0^{S_\infty}; \vdash \phi\}$. So we can generate all derivations of the sequent calculus of length n on the first n formulas and the first n terms using sequents of length at most n, and list any sentences found to be derivable. Do this for $n \in \mathbb{N}$ one at a time.

Theorem 130. *Let $\mathcal{A}$ be a finite alphabet and let $W \subseteq \mathcal{A}^*$ be a set of words over $\mathcal{A}$. If W is decidable then W is enumerable.*

Proof. Suppose W is decidable. Let $\mathfrak{P}$ be a decision procedure for W. To list W, start with a list for $\mathcal{A}^*$ (as described in the example above). For each word $x \in \mathcal{A}^*$, use $\mathfrak{P}$ to check whether $x \in W$ or $x \notin W$. If it is, then print x and continue. If it isn't, then continue with the next word in $\mathcal{A}^*$. ∎

Theorem 131. *Let $\mathcal{A}$ be a finite alphabet and let $W \subseteq \mathcal{A}^*$ be a set of words over $\mathcal{A}$. Then W is decidable if and only if W and $\mathcal{A}^* \setminus W$ are enumerable.*

Proof. Suppose first that W is decidable. We just saw that this means that W is enumerable, so we need only show that $\mathcal{A}^* \setminus W$ is enumerable. But since W is decidable, we have that $\mathcal{A}^* \setminus W$ is decidable. Indeed, given a decision procedure $\mathfrak{P}$ for W, to check if $x \in \mathcal{A}^*$ is a member of $\mathcal{A}^* \setminus W$, we simply run $\mathfrak{P}$ on x and check whether $x \notin W$ or not. So by the previous theorem again, we have that $\mathcal{A}^* \setminus W$ is enumerable.

Conversely, suppose that W and $\mathcal{A}^* \setminus W$ are both enumerable. So there are enumeration procedures $\mathfrak{P}$ and $\mathfrak{Q}$ for W and $\mathcal{A}^* \setminus W$ respectively. To decide whether $x \in W$, run each of $\mathfrak{P}$ and $\mathfrak{Q}$ on x (alternating one step at a time). At some point x will be printed by either $\mathfrak{P}$ or $\mathfrak{Q}$. If it is printed by $\mathfrak{P}$ then $x \in W$, and if it is printed by $\mathfrak{Q}$ then $x \notin W$. ∎

Definition 132. Let $\mathcal{A}$ and $\mathcal{B}$ be finite alphabets and let $f : \mathcal{A}^* \to \mathcal{B}^*$ be a function. We say f is *computable* if there exists a procedure that on every input $x \in \mathcal{A}^*$ halts having printed exactly one output $f(x) \in \mathcal{B}^*$.

10.2 Register Machines

Definition 133. Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ be a finite alphabet. We will let $R_0, R_1, \ldots$ denote registers. At each stage of a computation, each register will contain a word from $\mathcal{A}^*$. An *instruction* is something of the following form:

(1)
$$\mathrm{L}\ \mathrm{LET}\ R_i = R_i + a_j$$
for $L, i \in \mathbb{N}$ and $j \in \{0, \ldots, r\}$. This means add the letter a_j at the end of the word currently in register R_i.

(2)
$$\mathrm{L}\ \mathrm{LET}\ R_i = R_i - a_j$$
for $L, i \in \mathbb{N}$ and $j \in \{0, \ldots, r\}$. This means remove the last letter from the word in register R_i if it is a_j.

(3)
$$\mathrm{L}\ \mathrm{IF}\ R_i = \square\ \mathrm{THEN}\ \mathrm{L}'\ \mathrm{ELSE}\ \mathrm{L}_0\ \mathrm{OR} \cdots \mathrm{OR}\ \mathrm{L}_r$$
for $L, L', L_0, \ldots, L_r, i \in \mathbb{N}$. This means move to line L' if register R_i contains the empty word, and move to line L_ℓ if register R_i contains a word which ends with a_ℓ.

(4)
$$\mathrm{L}\ \mathrm{PRINT}$$
for $L \in \mathbb{N}$. This means print the word in register R_0.

(5)
$$\mathrm{L}\ \mathrm{HALT}$$
for $L \in \mathbb{N}$. This means end/stop/halt the computation.

For an instruction, the number L at the beginning is called its *label*. A *register program* (or just *program*) over $\mathcal{A}$ is a finite sequence $\alpha_0, \ldots, \alpha_k$ of instructions such that:

(i) Instruction α_i has label i;

(ii) For any instruction of the form (3), all of $L', L_0, \ldots, L_r$ are numbers in the range $\{0, \ldots, k\}$;

(iii) The last instruction α_k is

$$k \text{ HALT},$$

and no other instructions are of the form (5).

REMARK **134.** Each register program gives rise to a procedure. Indeed, simply "run" the list of instructions, altering the contents of the registers as necessary.

As a program is a finite sequence of instructions, it can only mention finitely many registers. So for a given program, there are only finitely many registers in play.

DEFINITION **135.** Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ be a finite alphabet, let $\mathbf{P}$ be a register program, and let $x, y \in \mathcal{A}^*$ be words over $\mathcal{A}$. We say $\mathbf{P}$ is *started with* x if $\mathbf{P}$ begins its computation with x in register R_0 and the empty word $\square$ in the other registers.

We write $\mathbf{P} : x \to \text{halt}$ if $\mathbf{P}$ started with x eventually reaches the halt instruction. We write $\mathbf{P} : x \to \infty$ if we do not have $\mathbf{P} : x \to \text{halt}$. We write $\mathbf{P} : x \to y$ if $\mathbf{P}$ started with x eventually halts, having printed exactly one output y.

REMARK **136.** If we ever write an instruction as

$$L \text{ GOTO } L',$$

then this should be understood as an abbreviation of

$$L \text{ IF } R_0 = \square \text{ THEN } L' \text{ ELSE } L' \text{ OR } \cdots \text{ OR } L'.$$

DEFINITION **137.** Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ be a finite alphabet, let $W \subseteq \mathcal{A}^*$ be a set of words over $\mathcal{A}$, and let $\mathbf{P}$ be a register program. We say $\mathbf{P}$ *decides* W if for all $x \in \mathcal{A}^*$ we have $\mathbf{P} : x \to \square$ if $x \in W$ and $\mathbf{P} : x \to y$ for some $y \neq \square$ if $x \notin W$.

We say W is *register decidable* (or *R-decidable*) if there is a register program which decides W.

DEFINITION **138.** Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ be a finite alphabet, let $W \subseteq \mathcal{A}^*$ be a set of words over $\mathcal{A}$, and let $\mathbf{P}$ be a register program. We say $\mathbf{P}$ *enumerates* W if $\mathbf{P}$ started with $\square$ prints out exactly the words in W (possibly with repetition and possibly with/without halting).

We say W is *register enumerable* (or *R-enumerable*) if there is a register program which enumerates W.

DEFINITION **139.** Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ and $\mathcal{B} = \{b_0, \ldots, b_s\}$ be finite alphabets, let $F : \mathcal{A}^* \to \mathcal{B}^*$ be a function, and let $\mathbf{P}$ be a register program over $\mathcal{A} \cup \mathcal{B} = \{a_0, \ldots, a_r, b_0, \ldots, b_s\}$. We say $\mathbf{P}$ *computes* F if for all $x \in \mathcal{A}^*$ we have $\mathbf{P} : x \to F(x)$.

We say F is *register computable* (or *R-computable*) if there is a register program which computes F.

Remark 140. [Church's Thesis] Recall that a register program gives rise to a procedure in the obvious way. It follows that if something is register decidable then it is also decidable. The same holds with 'decidable' replaced by 'enumerable' or 'computable'.

Church's Thesis states that for any reasonable model of computation, the converses of the above three statements also hold. This cannot be proved as we do not have a definition of 'reasonable', and more importantly, because we do not have a rigorous notion of 'procedure'. We will use Church's Thesis when convenient, assuming it to hold in our circumstances based on the experiences of others.

More generally, the Church Thesis (or the Church-Turing Thesis) says that our definition of 'computability' does indeed capture the notion we are attempting to define.

10.3 The Halting Problem for Register Machines

Definition 141. Let $\mathcal{A} = \{a_0, \dots, a_r\}$ be a finite alphabet and let $\mathbf{P}$ be a register program over $\mathcal{A}$. We define the alphabet

$$\mathcal{B}_\mathcal{A} := \{a_0, \dots, a_r, \mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{F}, \mathsf{G}, \mathsf{H}, \mathsf{I}, \mathsf{J}, \mathsf{K}, \mathsf{L}, \mathsf{M}, \mathsf{N}, \mathsf{O}, \mathsf{P}, \mathsf{Q}, \mathsf{R}, \mathsf{S}, \mathsf{T}, \mathsf{U}, \mathsf{V}, \mathsf{W}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, =, +, -, \square, \S\}.$$

To $\mathbf{P}$ we associate a word over $\mathcal{B}_\mathcal{A}$, by writing out the program explicitly and using § to separate lines. We could of course define this more rigorously but the idea should be clear with an example.

Example 142. The register program

$$\begin{aligned}&0 \text{ LET } R_1 = R_1 + a_0\\ &1 \text{ PRINT}\\ &2 \text{ HALT}\end{aligned}$$

becomes the word

$$0\mathsf{LETR}1 = \mathsf{R}1 + a_0\S 1\mathsf{PRINT}\S 2\mathsf{HALT}.$$

Definition 143. Let $\mathcal{A} = \{a_0, \dots, a_r\}$ be a finite alphabet and let $\mathbf{P}$ be a register program over $\mathcal{A}$. We can order $\mathcal{B}_\mathcal{A}$ by the lexicographical order $<_L$. This gives a natural bijection from $\mathbb{N}$ to $\mathcal{B}^*$. So there exists an $n \in \mathbb{N}$ such that the word corresponding to $\mathbf{P}$ is the image of n under this bijection. We define the *Gödel number* of $\mathbf{P}$ to be

$$\xi_\mathbf{P} := \underbrace{a_0 \cdots a_0}_{n \text{ times}}.$$

DEFINITION **144.** Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ be a finite alphabet. We let

$$\Pi := \{\xi_{\mathbf{P}}; \mathbf{P} \text{ is a register program over } \mathcal{A}\},$$

$$\Pi'_{\text{halt}} := \{\xi_{\mathbf{P}}; \mathbf{P} \text{ is a register program over } \mathcal{A} \text{ with } \mathbf{P} : \xi_{\mathbf{P}} \to \text{halt}\},$$

and

$$\Pi_{\text{halt}} := \{\xi_{\mathbf{P}}; \mathbf{P} \text{ is a register program over } \mathcal{A} \text{ with } \mathbf{P} : \square \to \text{halt}\}.$$

LEMMA **145.** *Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ be a finite alphabet. Then Π is register decidable.*

Proof. Let $x \in \mathcal{A}^*$. One can check if x is purely composed of a_0's. If it isn't, then $x \notin \Pi$. If it is, then it is composed on exactly n a_0's, for some $n \in \mathbb{N}$. Generate the n^{th} word of ${\mathcal{B}_{\mathcal{A}}}^*$ relative to the lexicographical ordering. Check if this represents a valid register program. If it does then $x \in \Pi$. Otherwise $x \notin \Pi$. ∎

THEOREM **146.** [UNDECIDABILITY OF THE HALTING PROBLEM] *Let $\mathcal{A} = \{a_0, \ldots, a_r\}$ be a finite alphabet.*

(a) The set Π'_{halt} is not register decidable.

(b) The set Π_{halt} is not register decidable.

Proof. First we prove part (a). Assume, for a contradiction, that there is a program $\mathbf{P}_0$ which decides Π'_{halt}. This means that for any program $\mathbf{P}$, we have

$$\mathbf{P}_0 : \xi_{\mathbf{P}} \to \square \text{ if } \mathbf{P} : \xi_{\mathbf{P}} \to \text{halt}\,,$$
$$\mathbf{P}_0 : \xi_{\mathbf{P}} \to y \text{ for some } y \neq \square \text{ if } \mathbf{P} : \xi_{\mathbf{P}} \to \infty.$$

We now construct a program $\mathbf{P}_1$ based on $\mathbf{P}_0$. To define $\mathbf{P}_1$, replace the last instruction

$$k \text{ HALT}$$

in $\mathbf{P}_0$ by

$$k \text{ IF } R_0 = \square \text{ THEN } k \text{ ELSE } k+1 \text{ OR } \cdots \text{ OR } k+1,$$

and replace all instructions of the form

$$L \text{ PRINT}$$

by

$$L \text{ GOTO } k.$$

Finally, add the line

$$k+1 \text{ HALT}$$

and increment all labels appropriately.

Note that

$$\mathbf{P}_1 : \xi_{\mathbf{P}} \to \infty \text{ if } \mathbf{P} : \xi_{\mathbf{P}} \to \text{halt}\,,$$
$$\mathbf{P}_1 : \xi_{\mathbf{P}} \to \text{halt if } \mathbf{P} : \xi_{\mathbf{P}} \to \infty.$$

Indeed, if $\mathbf{P} : \xi_{\mathbf{P}} \to$ halt then $\mathbf{P}_0 : \xi_{\mathbf{P}} \to \square$, so we must have gotten to a print instruction in $\mathbf{P}_0$, which we replaced with a jump into an infinite loop in $\mathbf{P}_1$, and similarly for the other case.

But now we see that

$$\mathbf{P}_1 : \xi_{\mathbf{P}_1} \to \infty \text{ if } \mathbf{P}_1 : \xi_{\mathbf{P}_1} \to \text{halt}\,,$$
$$\mathbf{P}_1 : \xi_{\mathbf{P}_1} \to \text{halt if } \mathbf{P}_1 : \xi_{\mathbf{P}_1} \to \infty,$$

which is of course absurd. Thus no program decides Π'_{halt} and it is not register decidable.

Now we turn to part (b). First we give a procedure that, given a program $\mathbf{P}$, gives another program $\mathbf{P}^+$ such that

$$\xi_{\mathbf{P}} \in \Pi'_{\text{halt}} \text{ if and only if } \xi_{\mathbf{P}^+} \in \Pi_{\text{halt}}. \tag{3}$$

Given $\mathbf{P}$, compute $\xi_{\mathbf{P}} = a_0 \cdots a_0$ (n times). Let $\mathbf{P}^+$ be the program that begins with the lines

$$\begin{gathered} 0 \text{ LET } R_0 = R_0 + a_0 \\ \vdots \\ n-1 \text{ LET } R_0 = R_0 + a_0 \end{gathered}$$

followed by the lines of $\mathbf{P}$ with all labels increased by n. Clearly $\mathbf{P}^+$ has the required property (3).

Now assume, for a contradiction, that we had a program which decides Π_{halt}. We will use this to describe a program which decides Π'_{halt}. Given $x \in \mathcal{A}^*$, first decide whether or not $x \in \Pi$. If $x \notin \Pi$ then $x \notin \Pi'_{\text{halt}}$. Otherwise, $x = \xi_{\mathbf{P}}$ for some program $\mathbf{P}$, and we can compute which program this is. Now we can compute $\mathbf{P}^+$ and $\xi_{\mathbf{P}^+}$. Now, using our assumption, decide whether or not $\xi_{\mathbf{P}^+} \in \Phi_{\text{halt}}$, and hence whether or not $\xi_{\mathbf{P}} \in \Pi'_{\text{halt}}$. But this contradicts part (a), so there is no such program and thus Π_{halt} is not decidable. ∎

Lemma 147. *The set* Π_{halt} *is enumerable.*

Proof. To enumerate Π_{halt}, for each $n \in \mathbb{N}$ with $n \geq 1$, generate the finitely many programs with Gödel number of length at most n and then run each of these programs with input $\square$ for up to n steps. List out all programs which halted. ∎

Corollary 148. *The set* $\mathcal{A}^* \setminus \Pi_{\text{halt}}$ *is not enumerable.*

Proof. Assume, for a contradiction, that $\mathcal{A}^* \setminus \Pi_{\text{halt}}$ is enumerable. Since Π_{halt} is enumerable, we saw that this means Π_{halt} is decidable. This contradicts the previous theorem. ∎

10.4 The Undecidability of First-Order Logic

10.5 Trahtenbrot's Theorem and the Incompleteness of Second-Order Logic

10.6 Theories and Decidability

Definition 149. Let S be a first-order language and let $T \subseteq L_0^S$ be a set of S-sentences. We say that T is a *theory* if T is satisfiable and whenever $\phi \in L_0^S$ is such that $T \models \phi$, we have $\phi \in T$ (that is, T is closed under logical consequence).

Example 150. For any S-structure $\mathfrak{A}$, the theory of $\mathfrak{A}$, $\mathrm{Th}(\mathfrak{A})$, is a theory.

Definition 151. Let S be a first-order language and let $\Phi \subseteq L_0^S$ be a set of S-sentences. We let

$$\Phi^{\models} := \{\phi \in L_0^S ; \Phi \models \phi\}.$$

Proposition 152. *Let S be a first-order language and let $\Phi \subseteq L_0^S$ be a set of S-sentences. If Φ is satisfiable then $\Phi^{\models}$ is a theory, and if Φ is a theory then $\Phi^{\models} = \Phi$.*

Proof. Suppose $\mathrm{Sat}\,\Phi$. So there exists an S-structure $\mathfrak{A}$ such that $\mathfrak{A} \models \Phi$. Suppose $\phi \in \Phi^{\models}$. This means $\Phi \models \phi$. As $\mathfrak{A} \models \Phi$, we see $\mathfrak{A} \models \phi$. Thus $\mathfrak{A} \models \Phi^{\models}$ and so $\mathrm{Sat}\,\Phi^{\models}$. Now if $\Phi^{\models} \models \phi$ for some $\phi \in L_0^S$, then $\Phi \models \phi$ (indeed, if $\mathfrak{I} \models \Phi$ for an S-interpretation $\mathfrak{I}$, then $\mathfrak{I} \models \Phi^{\models}$ and hence $\mathfrak{I} \models \phi$ by assumption) and so $\phi \in \Phi^{\models}$. So $\Phi^{\models}$ is closed under logical consequence and is thus a theory.

Now suppose that Φ is a theory. Certainly $\Phi \subseteq \Phi^{\models}$. Suppose $\phi \in \Phi^{\models}$. This means $\Phi \models \phi$. But Φ is closed under logical consequence, so $\phi \in \Phi$. Hence $\Phi^{\models} \subseteq \Phi$ and thus equality holds. ∎

Definition 153. Let Φ_{PA} consist of exactly the following S_{ar}-sentences:

$\forall v_0 \neg + v_0 1 \equiv 0$;

$\forall v_0 + v_0 0 \equiv v_0$;

$\forall v_0 \cdot v_0 0 \equiv 0$;

$\forall v_0 \forall v_1 (+v_0 1 \equiv +v_1 1 \rightarrow v_0 \equiv v_1)$;

$\forall v_0 \forall v_1 + v_0 + v_1 1 \equiv + + v_0 v_1 1$;

$\forall v_0 \forall v_1 \cdot v_0 + v_1 1 \equiv + \cdot v_0 v_1 v_0$;

for all variables $x_1, \ldots, x_n, y$ and all $\phi \in L^{S_{ar}}$ such that $\mathrm{free}_{S_{ar}}(\phi) \subseteq \{x_1, \ldots, x_n, y\}$, the sentence

$$\forall x_1 \cdots \forall x_n ((\phi\tfrac{0}{y} \wedge \forall y(\phi \rightarrow \phi\tfrac{+y1}{y})) \rightarrow \forall y \phi).$$

Remark 154. Note that $\mathfrak{N} \models \Phi_{PA}$, so $\operatorname{Sat}\Phi_{PA}$. This means that $\Phi_{PA}^{\models}$ is a theory, called first-order Peano arithmetic. This also means that $\Phi_{PA}^{\models} \subseteq \mathrm{Th}(\mathfrak{N})$. We will in fact see that $\Phi_{PA}^{\models} \subset \mathrm{Th}(\mathfrak{N})$.

Definition 155. Let S be a finite first-order language and let $T \subseteq L_0^S$ be a theory. We say T is *axiomatizable* if there is a decidable set $\Phi \subseteq L_0^S$ of S-sentences such that $T = \Phi^{\models}$. We say T is *finitely axiomatizable* if there is a finite set $\Phi \subseteq L_0^S$ of S-sentences such that $T = \Phi^{\models}$.

Remark 156. Any finite set is decidable, so a finitely axiomatizable theory is axiomatizable.

Theorem 157. *Let S be a finite first-order language and let $T \subseteq L_0^S$ be a theory. If T is axiomatizable then T is enumerable.*

Proof. Since T is axiomatizable, there exists a decidable set $\Phi \subseteq L_0^S$ of S-sentences such that $T = \Phi^{\models}$. Generate systematically all derivable sequents and check in each case whether the members of the antecedent belong to Φ (since it is decidable). If yes and the succedent is a sentence, then output it. ∎

Remark 158. It is not the case that an axiomatizable theory is necessarily decidable. The set of all S^{∞}-sentences is a counter example. See book for details.

Definition 159. Let S be a first-order language and let $T \subseteq L_0^S$ be a theory. We say that T is *complete* if for every S-sentence $\phi \in L_0^S$ we have $\phi \in T$ or $\neg\phi \in T$.

Example 160. For any structure $\mathfrak{A}$, $\mathrm{Th}(\mathfrak{A})$ is a complete theory.

Theorem 161. *Let S be a finite first-order language and let $T \subseteq L_0^S$ be a theory.*

(a) If T is axiomatizable and complete then T is decidable.

(b) If T is enumerable and complete then T is decidable.

Proof. The previous theorem showed that if T is axiomatizable then T is enumerable, so we need only prove part (b). For any S-sentence $\phi \in L_0^S$, since T is complete, $\phi \in T$ or $\neg\phi \in T$. Since T is consistent (as it is a theory), we cannot have both $\phi \in T$ and $\neg\phi \in T$. Hence exactly one of ϕ and $\neg\phi$ is in T.

To decide whether $\phi \in T$, enumerate T until either ϕ or $\neg\phi$ appears. As we just argued, one of them is in T and so will eventually appear. If ϕ appears then $\phi \in T$, and if $\neg\phi$ appears then $\phi \notin T$. Since they are not both in T, we will not produce the wrong answer. ∎

DEFINITION **162.** Let $\mathbf{P}$ be a register program over $\{|\}$ with k instructions such that R_n is the largest register mentioned by the instructions of $\mathbf{P}$, and let $(L, m_0, \dots, m_n) \in \mathbb{N}^{n+2}$ be such that $L \le k$. We say that $(L, m_0, \dots, m_n)$ *is the configuration of* $\mathbf{P}$ *after* s *steps* if $\mathbf{P}$ started with $\square$ runs for at least s steps, and after s steps, instruction L is to be executed next, with registers $R_0, \dots, R_n$ containing $m_0, \dots, m_n$ respectively (by which we mean $|\cdots|$ m_i times).

THEOREM **163.** [UNDECIDABILITY OF ARITHMETIC] *The theory of arithmetic,* $\mathrm{Th}(\mathfrak{N})$, *is not decidable.*

Proof. We will effectively assign, for each program $\mathbf{P}$ over $\{|\}$, an S_{ar}-sentence $\phi_{\mathbf{P}}$ such that $\mathfrak{N} \models \phi_{\mathbf{P}}$ if and only if $\mathbf{P} : \square \to \text{halt}$. Indeed, assume we have this and assume, for a contradiction, that $\mathrm{Th}(\mathfrak{N})$ is decidable. Then to decide whether or not a program $\mathbf{P}$ over $\{|\}$ is in Π_{halt}, we can compute $\phi_{\mathbf{P}}$ and then decide whether or not $\phi_{\mathbf{P}}$ is in $\mathrm{Th}(\mathfrak{N})$, which would decide the halting set Π_{halt}. This is absurd though, so it suffices to prove this claim.

So we effectively construct $\phi_{\mathbf{P}}$. Let k be the label of the last line in $\mathbf{P}$, and let R_n be the biggest register mentioned by the instructions of $\mathbf{P}$. By lemma (164), there exists an effectively constructed S_{ar}-formula $\chi_{\mathbf{P}} \in L^S_{2n+3}$ such that $\mathfrak{N} \models \chi_{\mathbf{P}}[\ell_0, \dots, \ell_n, L, m_0, \dots, m_n]$ if and only if $\mathbf{P}$, started with configuration $(0, \ell_0, \dots, \ell_n)$, after finitely many steps reaches configuration $(L, m_0, \dots, m_n)$. Now we let

$$\phi_{\mathbf{P}} := \exists v_0 \cdots \exists v_n \chi_{\mathbf{P}}(\underline{0}, \dots, \underline{0}, \underline{k}, v_0, \dots, v_n).$$

Since the only way program $\mathbf{P}$ can halt is to reach a configuration of the form $(k, m_0, \dots, m_n)$ after finitely many steps, it is easy to see that $\mathfrak{N} \models \phi_{\mathbf{P}}$ if and only if $\mathbf{P}$ started with input $\square$ halts. This completes the proof. ∎

LEMMA **164.** *This is the* $\chi_{\mathbf{P}}$*-lemma, which we have omitted — see class notes.*

Proof. Omitted. ∎

LEMMA **165.** [β-FUNCTION LEMMA] *This is the* β*-function lemma, which we have omitted — see class notes.*

Proof. Omitted. ∎

COROLLARY **166.** *The theory of arithmetic,* $\mathrm{Th}(\mathfrak{N})$, *is neither axiomatizable nor enumerable. In particular,* $\Phi_{PA}^{\models} \subset \mathrm{Th}(\mathfrak{N})$.

Proof. We saw that if a theory is complete and either axiomatizable or enumerable then it is decidable, but we just saw that $\mathrm{Th}(\mathfrak{N})$ is not decidable and we know that it is complete. Moreover, since $\Phi_{PA}^{\models}$ is axiomatizable (by Φ_{PA}) and since $\mathrm{Th}(\mathfrak{N})$ is not axiomatizable, they cannot be equal. But we said $\Phi_{PA}^{\models} \subseteq \mathrm{Th}(\mathfrak{N})$, and thus we conclude $\Phi_{PA}^{\models} \subset \mathrm{Th}(\mathfrak{N})$. ∎

Definition 167. Let $r \in \mathbb{N}$, let $R \subseteq \mathbb{N}^r$ be an r-ary relation, and let $\mathbf{P}$ be a program over $\{|\}$. We say $\mathbf{P}$ *decides* R if $\mathbf{P}$ started with configuration $(0, \ell_0, \dots, \ell_{r-1})$ eventually halts, producing $\square$ if $(\ell_0, \dots, \ell_{r-1}) \in R$ and producing $\underline{n} \neq 0$ otherwise. We say R is *decidable* if there is a program over $\{|\}$ which decides R.

Definition 168. Let $r \in \mathbb{N}$, let $F : \mathbb{N}^r \to \mathbb{N}$ be an r-ary function, and let $\mathbf{P}$ be a program over $\{|\}$. We say $\mathbf{P}$ *computes* F if $\mathbf{P}$ started with configuration $(0, \ell_0, \dots, \ell_{r-1})$ eventually halts, producing $F(\ell_0, \dots, \ell_{r-1})$. We say F is *computable* if there is a program over $\{|\}$ which computes F.

Theorem 169. *Let $r \in \mathbb{N}$, let $R \subseteq \mathbb{N}^r$ be an r-ary relation, and let $f : \mathbb{N}^r \to \mathbb{N}$ be an r-ary function.*

(a) If R is decidable then there is an S_{ar}-formula $\phi \in L^S_r$ such that for all $(\ell_0, \dots, \ell_{r-1}) \in \mathbb{N}^r$,

$$(\ell_0, \dots, \ell_{r-1}) \in R \text{ if and only if } \mathfrak{N} \models \phi[\underline{\ell_0}, \dots, \underline{\ell_{r-1}}].$$

(b) If f is computable then there is an S_{ar}-formula $\phi \in L^S_{r+1}$ such that for all $(\ell_0, \dots, \ell_r) \in \mathbb{N}^{r+1}$,

$$f(\ell_0, \dots, \ell_{r-1}) = \ell_r \text{ if and only if } \mathfrak{N} \models \phi[\underline{\ell_0}, \dots, \underline{\ell_r}],$$

and in particular

$$\mathfrak{N} \models \exists^{=1} x \phi(\underline{\ell_0}, \dots, \underline{\ell_{r-1}}, x).$$

Proof. We will prove (a) and leave (b) as a similar exercise. Let $\mathbf{P}$ be a program which decides R. Let $n \geq r-1$ be such that no register larger than R_n is mentioned in the instructions of $\mathbf{P}$. Let $\alpha_{L_0}, \dots, \alpha_{L_m}$ be the print instructions in $\mathbf{P}$. Now we note that $(\ell_0, \dots, \ell_{r-1}) \in R$ if and only if $\mathbf{P}$, beginning with configuration $(0, \ell_0, \dots, \ell_{r-1}, 0, \dots, 0)$, after finitely many steps reaches a configuration of the form $(L_i, 0, m_1, \dots, m_n)$ for some $i \in \{0, \dots, m\}$. This is because this is the only way to print $\square$; arrive at a print instruction with 0 in register R_0. But this happens if and only if $\mathfrak{N} \models \exists x_1 \cdots \exists x_n \bigvee_{i=0}^m \chi_{\mathbf{P}}(\underline{\ell_0}, \dots, \underline{\ell_{r-1}}, 0, \dots, 0, \underline{L_i}, 0, x_1, \dots, x_n)$, where $\chi_{\mathbf{P}}$ is from lemma (164). So we let

$$\phi := \exists x_1 \cdots \exists x_n \bigvee_{i=0}^{m} \chi_{\mathbf{P}}(v_0, \dots, v_{r-1}, 0, \dots, 0, \underline{L_i}, 0, \dots, 0).$$

It follows quickly from our discussion above that ϕ has the desired properties. ∎

10.7 Self-Referential Statements and Gödel's Incompleteness Theorems

DEFINITION **170.** Let $\Phi \subseteq L_0^{S_{ar}}$, let $r \in \mathbb{N}$, let $R \subseteq \mathbb{N}^r$ be an r-ary relation, and let $\phi \in L_r^{S_{ar}}$. We say ϕ *represents* R in Φ if for all $(n_0, \ldots, n_{r-1}) \in \mathbb{N}^r$,

$$\text{if } (n_0, \ldots, n_{r-1}) \in R \text{ then } \Phi \models \phi(\underline{n_0}, \ldots, \underline{n_{r-1}}),$$
$$\text{if } (n_0, \ldots, n_{r-1}) \in R \text{ then } \Phi \models \neg\phi(\underline{n_0}, \ldots, \underline{n_{r-1}}).$$

We say R is *representable* in Φ if there exists a formula which represents R in Φ.

DEFINITION **171.** Let $\Phi \subseteq L_0^{S_{ar}}$, let $r \in \mathbb{N}$, let $F : \mathbb{N}^r \to \mathbb{N}$ be an r-ary function, and let $\phi \in L_{r+1}^{S_{ar}}$. We say ϕ *represents* F in Φ if for all $(n_0, \ldots, n_r) \in \mathbb{N}^{r+1}$,

$$\text{if } F(n_0, \ldots, n_{r-1}) = n_r \text{ then } \Phi \vdash \phi(\underline{n_0}, \ldots, \underline{n_r}),$$
$$\text{if } F(n_0, \ldots, n_{r-1}) \neq n_r \text{ then } \Phi \vdash \neg\phi(\underline{n_0}, \ldots, \underline{n_r}),$$
$$\Phi \vdash \exists^{=1} x \phi(\underline{n_0}, \ldots, \underline{n_{r-1}}, x).$$

We say F is *representable* in Φ if there exists a formula which represents F in Φ.

LEMMA **172.** *Let $\Phi, \Psi \subseteq L_0^{S_{ar}}$.*

(a) If Φ is inconsistent then every relation over $\mathbb{N}$ and every function over $\mathbb{N}$ is representable in Φ.

(b) If $\Phi \subseteq \Psi$ then the relations and functions over $\mathbb{N}$ which are representable in Φ are also representable in Ψ.

(c) If Φ is consistent and decidable then every relation representable in Φ is decidable and every function representable in Φ is computable.

Proof. (a) We can take $0 \equiv 0$ to represent any relation or function over $\mathbb{N}$. Indeed, since Φ is inconsistent, $\Phi \vdash \phi$ for all $\phi \in L^{S_{ar}}$. Hence the definition is clearly satisfied.

(b) If ϕ represents a relation or function in Φ, then ϕ also represents the same relation or function in Ψ. Indeed, if $\Phi \models \psi$ then $\Psi \models \psi$ for any $\psi \in L^{S_{ar}}$.

(c) Suppose $r \in \mathbb{N}$ and $R \subseteq \mathbb{N}^r$ is an r-ary relation over $\mathbb{N}$. Suppose $\phi \in L_r^S$ represents R in Φ. Let $(n_0, \ldots, n_{r-1}) \in \mathbb{N}^r$. To decide whether or not $(n_0, \ldots, n_{r-1}) \in R$, note that $\Phi^{\vdash}$ is enumerable, since Φ is decidable. So we can enumerate $\Phi^{\vdash}$ until one of $\phi(\underline{n_0}, \ldots, \underline{n_{r-1}})$ or $\neg\phi(\underline{n_0}, \ldots, \underline{n_{r-1}})$ appears. Since ϕ represents R in Φ, one of these must appear by definition. Since Φ is consistent, both cannot appear. Thus exactly one will appear and this tells us whether $(n_0, \ldots, n_{r-1}) \in R$.

Suppose $r \in \mathbb{N}$ and $F : \mathbb{N}^r \to \mathbb{N}$ is an r-ary function over $\mathbb{N}$. Suppose $\phi \in L^S_{r+1}$ represents F in Φ. Let $(n_0, \dots, n_{r-1}) \in \mathbb{N}^r$. To compute $F(n_0, \dots, n_{r-1})$, again note that $\Phi^\vdash$ is enumerable. Enumerate $\Phi^\vdash$ until a formula of the form $\phi(\underline{n_0}, \dots, \underline{n_{r-1}}, \underline{m})$ appears. Since Φ is consistent and $\Phi \vDash \exists^{=1} x \phi(\underline{n_0}, \dots, \underline{n_{r-1}}, x)$, exactly one such formula will appear. This tells us $F(n_0, \dots, n_{r-1}) = m$.

∎

Definition **173.** Let $\Phi \subseteq L_0^{S_{ar}}$. We say Φ *allows representations* if all decidable relations over $\mathbb{N}$ and all computable functions over $\mathbb{N}$ are representable in Φ.

Theorem **174.** *The theory of arithmetic,* $\mathrm{Th}(\mathfrak{N})$, *allows representations.*

Proof. Note that $\mathrm{Th}(\mathfrak{N})$ is a complete theory. It is an easy exercise to observe that theorem (169) directly implies the desired result (because it is complete). ∎

Theorem **175.** *The set* Φ_{PA} *allows representations.*

Proof. Omitted. ∎

Remark **176.** To prove the previous theorem, we would proceed as in the proof that the theory of arithmetic allows representations, which uses lemma (164). Hence it is required to create an analogous lemma in which $\mathrm{Th}(\mathfrak{N})$ is replaced by Φ_{PA}.

Remark **177.** For the remainder of this section, we fix an effective coding of the S_{ar}-formulas into $\mathbb{N}$ (called a *Gödel numbering*) which is surjective. For an S_{ar}-formula $\phi \in L^{S_{ar}}$ we will write n^ϕ for the Gödel number of ϕ.

The goal here is to examine self-referential formulas. Clearly a S_{ar}-formula ϕ cannot contain itself as a sub-formula, so it cannot directly talk about itself. But ϕ can mention its Gödel number, n^ϕ. It is in this sense that a formula can be self-referential.

Theorem **178.** [Fixed-Point Theorem] *Let* $\Phi \subseteq L_0^{S_{ar}}$ *and let* $\psi \in L_1^{S_{ar}}$. *If* Φ *allows representations then there exists a* $\phi \in L_0^{S_{ar}}$ *such that*

$$\Phi \vdash (\phi \leftrightarrow \psi(\underline{n^\phi})).$$

Proof. Define the function $F : \mathbb{N}^2 \to \mathbb{N}$ by

$$F(k, m) := \begin{cases} n^{\chi(\underline{m})} & \text{if } k = n^\chi \text{ for some } \chi \in L_1^{S_{ar}} \\ 0 & \text{otherwise.} \end{cases}$$

Note that F is clearly computable and that $F(n^\chi, m) = n^{\chi(\underline{m})}$ for any $\chi \in L_1^{S_{ar}}$.

Since Φ allows representations, there exists an $\alpha \in L_3^{S_{ar}}$ such that for all $k, m, t \in \mathbb{N}$ we have

$$\begin{aligned} &\text{if } F(k,m) = t \text{ then } \Phi \vdash \alpha(\underline{k}, \underline{m}, \underline{t}), \\ &\text{if } F(k,m) \neq t \text{ then } \Phi \vdash \neg\alpha(\underline{k}, \underline{m}, \underline{t}), \\ &\Phi \vdash \exists^{=1} z \alpha(\underline{k}, \underline{m}, \underline{z}). \end{aligned}$$

Let

$$\beta := \forall z(\alpha(x, x, z) \to \psi(z))$$

and take

$$\phi := \forall z(\alpha(\underline{n^\beta}, \underline{n^\beta}, z) \to \psi(z)).$$

We claim that $\Phi \vdash (\phi \leftrightarrow \psi(\underline{n^\phi}))$. First we show that $\Phi \vdash (\phi \to \psi(\underline{n^\phi}))$. Well, note that $\beta \in L_1^{S_{ar}}$, so $F(n^\beta, n^\beta) = n^{\beta(\underline{n^\beta})}$. But we see $\beta(\underline{n^\beta}) = \phi$, so this is actually $F(n^\beta, n^\beta) = n^\phi$. By definition of α this means

$$\Phi \vdash \alpha(\underline{n^\beta}, \underline{n^\beta}, \underline{n^\phi}). \tag{4}$$

Now it is clear by definition of ϕ that

$$\{\phi\} \vdash (\alpha(\underline{n^\beta}, \underline{n^\beta}, \underline{n^\phi}) \to \psi(\underline{n^\phi})).$$

So if $\Phi \vdash \phi$ then $\Phi \vdash \psi(\underline{n^\phi})$. In other words, $\Phi \vdash (\phi \to \psi(\underline{n^\phi}))$.

It remains to show that $\Phi \vdash (\psi(\underline{n^\phi}) \to \phi)$. By definition of α we see $\Phi \vdash \exists^{=1} z \alpha(\underline{n^\beta}, \underline{n^\beta}, z)$. Together with (4), we get

$$\Phi \vdash \forall z(\alpha(\underline{n^\beta}, \underline{n^\beta}, z) \to z \equiv \underline{n^\phi}).$$

Suppose $\Phi \vdash \psi(\underline{n^\phi})$. Then the above shows $\Phi \vdash \forall z(\alpha(\underline{n^\beta}, \underline{n^\beta}, z) \to \psi(z))$. However this is just the definition of ϕ, so we have $\Phi \vdash \phi$. That is to say, $\Phi \vdash (\psi(\underline{n^\phi}) \to \phi)$. ∎

Definition 179. Let $\Phi \subseteq L_0^{S_{ar}}$ and let $\Psi \subseteq L^{S_{ar}}$. We say Ψ is *representable* in Φ if the unary relation $\{(n^\phi); \phi \in \Psi\}$ over $\mathbb{N}$ is representable in Φ.

Lemma 180. *Let $\Phi \subseteq L_0^{S_{ar}}$. If Φ is consistent and allows representations then $\Phi^\vdash$ is not representable in Φ.*

Proof. Assume, for a contradiction, that $\Phi^\vdash$ is representable in Φ. This means that there exists a formula $\chi \in L_1^{S_{ar}}$ which represents $\{n^\phi; \phi \in \Phi^\vdash\}$ in Φ. In particular, for any $\alpha \in L_0^{S_{ar}}$ we have

$$\begin{aligned} &\text{if } \alpha \in \Phi^\vdash \text{ then } \Phi \vdash \chi(\underline{n}), \\ &\text{if } \alpha \notin \Phi^\vdash \text{ then } \Phi \vdash \neg\chi(\underline{n}). \end{aligned}$$

To put this another way, for any $\alpha \in L_0^{S_{ar}}$ we have

$$\text{if } \Phi \vdash \alpha \text{ then } \Phi \vdash \chi(\underline{n}^{\alpha}),$$
$$\text{if not } \Phi \vdash \alpha \text{ then } \Phi \vdash \neg\chi(\underline{n}^{\alpha}).$$

Since Φ is consistent, we also know

$$\text{if } \Phi \vdash \neg\chi(\underline{n}^{\alpha}) \text{ then not } \Phi \vdash \chi(\underline{n}^{\alpha})$$

and hence

$$\text{if } \Phi \vdash \neg\chi(\underline{n}^{\alpha}) \text{ then not } \Phi \vdash \alpha.$$

Combining everything we get

$$\text{not } \Phi \vdash \alpha \text{ if and only if } \Phi \vdash \neg\chi(\underline{n}^{\alpha})$$

for any $\alpha \in L_0^{S_{ar}}$.

But we can apply the fixed-point theorem to $\neg\chi$ to conclude that there is a formula $\phi \in L_0^{S_{ar}}$ such that

$$\Phi \vdash (\phi \leftrightarrow \neg\chi(\underline{n}^{\phi})).$$

Putting the last two equations together (with ϕ in place of α) yields

$$\Phi \vdash \phi \text{ if and only if } \Phi \vdash \neg\chi(\underline{n}^{\phi}) \text{ if and only if not } \Phi \vdash \phi.$$

Contradiction. ∎

THEOREM 181. [TARSKI'S THEOREM] *Let $\Phi \subseteq L_0^{S_{ar}}$. If Φ is consistent and allows representations then $\Phi^{\models}$ is not representable in Φ.*

In particular, $\mathrm{Th}(\mathfrak{N})$ *is not representable in* $\mathrm{Th}(\mathfrak{N})$.

Proof. Since $\Phi^{\models} = \Phi^{\vdash}$, the first part follows immediately form the previous lemma.

To see that the theory of arithmetic is not representable in itself, note that $\mathrm{Th}(\mathfrak{N})$ is consistent and allows representations, and that $\mathrm{Th}(\mathfrak{N})^{\models} = \mathrm{Th}(\mathfrak{N})$ because it is a theory. Therefore this is a direct consequence of the first part of the theorem. ∎

THEOREM 182. [GÖDEL'S FIRST INCOMPLETENESS THEOREM] *Let $\Phi \subseteq L_0^{S_{ar}}$. If Φ is consistent, decidable, and allows representations, then there is an S_{ar}-sentence $\phi \in L_0^{S_{ar}}$ such that neither $\Phi \vdash \phi$ nor $\Phi \vdash \neg\phi$.*

Proof. Assume, for a contradiction, that for every S_{ar}-sentence $\phi \in L_0^{S_{ar}}$, we have either $\Phi \vdash \phi$ or $\Phi \vdash \neg\phi$. Then $\Phi^{\vdash}$ is a complete theory (because Φ is consistent). Since Φ is decidable, $\Phi^{\vdash}$ is enumerable. But we saw that a complete enumerable theory was decidable. Thus $\Phi^{\vdash}$ is decidable. Since Φ allows representations, this means $\Phi^{\vdash}$ is representable in Φ. But because Φ is consistent and allows representations, the previous lemma says that $\Phi^{\vdash}$ is not representable in Φ. Contradiction. ∎

Definition 183. Let $\Phi \subseteq L_0^{S_{ar}}$ be decidable and allow representations. Fix some (any) effective enumeration of all derivations in the sequent calculus. We define a binary relation $H_\Phi \subseteq \mathbb{N}^2$ by $(t, m) \in H_\Phi$ if and only if the m^{th} derivation in the sequent calculus ends with a sequent of the form $\psi_0 \cdots \psi_{k-1} \phi$ with $\psi_0, \ldots, \psi_{k-1} \in \Phi$ and $t = n^\phi$.

Remark 184. Note that H_Φ is decidable since Φ is decidable. It is not hard to see that $\Phi \vdash \phi$ if and only if there is an $m \in \mathbb{N}$ such that $(n^\phi, m) \in H_\Phi$. Since Φ allows representations, there is some formula $\psi_\Phi \in L_2^{S_{ar}}$ which represents H_Φ in Φ. If we let $\mathrm{Der}_\Phi := \exists v_1 \psi_\Phi(v_0, v_1)$, then we can apply the fixed-point theorem to $\neg \mathrm{Der}_\Phi$ to get that there exists some $\phi_\Phi \in L_0^{S_{ar}}$ such that

$$\Phi \vdash (\phi_\Phi \leftrightarrow \neg \mathrm{Der}_\Phi(\underline{n^{\phi_\Phi}})).$$

Lemma 185. *Let $\Phi \subseteq L_0^{S_{ar}}$ be decidable and allow representations. If Φ is consistent then not $\Phi \vdash \phi_\Phi$.*

Proof. Assume, for a contradiction, that $\Phi \vdash \phi_\Phi$. Then there exists an $m \in \mathbb{N}$ such that $(n^{\phi_\Phi}, m) \in H_\Phi$. Since ψ_Φ represents H_Φ in Φ, we see $\Phi \vdash \psi_\Phi(\underline{n^{\phi_\Phi}}, \underline{m})$. That is to say, $\Phi \vdash \mathrm{Der}_\Phi(\underline{n^{\phi_\Phi}})$ (by definition of Der).

Now note that our original assumption, combined with the definition of ϕ_Φ, yields $\Phi \vdash \neg \mathrm{Der}_\Phi(\underline{n^{\phi_\Phi}})$. Thus Φ is inconsistent. Contradiction. ∎

Remark 186. Note that $\Phi \vdash 0 \equiv 0$, so we have $\mathrm{Con}\,\Phi$ if and only if not $\Phi \vdash \neg 0 \equiv 0$. We hence define

$$\mathrm{Consis}_\Phi := \neg \mathrm{Der}_\Phi(\underline{n^{\neg 0 \equiv 0}}).$$

For $\Phi \supseteq \Phi_{\mathrm{PA}}$, the previous lemma can be formalized as

$$\Phi \vdash (\mathrm{Consis}_\Phi \to \neg \mathrm{Der}_\Phi(\underline{n^{\phi_\Phi}})),$$

though it is relatively tedious. We assume that this has been done in the next theorem.

Theorem 187. [Gödel's Second Incompleteness Theorem] *Let $\Phi \subseteq L_0^{S_{ar}}$. If Φ is consistent and decidable, and $\Phi_{\mathrm{PA}} \subseteq \Phi$, then not $\Phi \vdash \mathrm{Consis}_\Phi$.*

Proof. Assume, for a contradiction, that $\Phi \vdash \mathrm{Consis}_\Phi$. Then by the unproven remark, we get $\Phi \vdash \neg \mathrm{Der}_\Phi(\underline{n^{\phi_\Phi}})$. But we saw that $\Phi \vdash (\phi_\Phi \leftrightarrow \neg \mathrm{Der}_\Phi(\underline{n^{\phi_\Phi}}))$, so this means $\Phi \vdash \phi_\Phi$. This contradicts the previous lemma. ∎

11 Free Models and Logic Programming

12 An Algebraic Characterization of Elementary Equivalence

12.1 Finite and Partial Isomorphisms

Definition 188. Let S be a first-order language, let $\mathfrak{A} = (A, \mathfrak{a})$ and $\mathfrak{B} = (B, \mathfrak{b})$ be S-structures, and let $p : A \to B$ be a partial function. We say p is a *partial isomorphism* from $\mathfrak{A}$ to $\mathfrak{B}$ if:

1. p is injective;
2. for every n-ary relation symbol $R \in S$ and all $a_1, \dots, a_n \in \mathrm{dom}(p)$,
$$(a_1, \dots, a_n) \in \mathfrak{a}(R) \text{ if and only if } (p(a_1), \dots, p(a_n)) \in \mathfrak{b}(R);$$
3. for every n-ary function symbol $f \in S$ and all $a, a_1, \dots, a_n \in \mathrm{dom}(p)$,
$$\mathfrak{a}(f)(a_1, \dots, a_n) = a \text{ if and only if } \mathfrak{b}(f)(p(a_1), \dots, p(a_n)) = p(a);$$
4. for every constant symbol $c \in S$ and all $a \in \mathrm{dom}(p)$,
$$\mathfrak{a}(c) = a \text{ if and only if } \mathfrak{b}(c) = p(a).$$

Definition 189. Let S be a first-order language and let $\mathfrak{A}$ and $\mathfrak{B}$ be S-structures. We let $\mathrm{Part}(\mathfrak{A}, \mathfrak{B})$ denote the set of all partial isomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$.

Example 190. The empty map is a partial isomorphism between any two S-structures, vacuously.

Example 191. If $\pi : \mathfrak{A} \cong \mathfrak{B}$ and $C \subseteq A$ then $\pi|_C \in \mathrm{Part}(\mathfrak{A}, \mathfrak{B})$. That is to say, the restriction of an isomorphism is a partial isomorphism. In particular, an isomorphism is a partial isomorphism.

Example 192. The partial map $p : \mathbb{R} \to \mathbb{Z}$ given by $p(2) := 2$ and $p(3) := 6$ is a partial isomorphism from $(\mathbb{R}, \mathfrak{a})$ to $(\mathbb{Z}, \mathfrak{b})$ (as $\{0, +\}$-structures with the standard $\mathfrak{a}$ and $\mathfrak{b}$). Indeed, there are no relation symbols to check, 2 and 3 can never be added to get back 2 or 3 and 2 and 6 can never be added to get back 2 or 6, and $\mathfrak{a}(0) = 0 \notin \mathrm{dom}(p)$ and $\mathfrak{b}(0) \notin \mathrm{range}(p)$.

On the other hand, the partial map $q : \mathbb{R} \to \mathbb{Z}$ given by $q(2) := 2$ and $q(3) := 4$ is not a partial isomorphism from $(\mathbb{R}, \mathfrak{a})$ to $(\mathbb{Z}, \mathfrak{b})$. Indeed, we have $q(2) + q(2) = 2 + 2 = 4 = q(3)$ while $2 + 2 = 4 \neq 3$.

Finally, the partial map $g : \mathbb{R} \to \mathbb{Z}$ given by $g(n) := n$ for all $n \in \mathbb{Z}$ is a partial isomorphism which is surjective but not an isomorphism.

Proposition **193.** *Let* S *be a relational first-order language, let* $\mathfrak{A} = (A, \mathfrak{a})$ *and* $\mathfrak{B} = (B, \mathfrak{b})$ *be* S*-structures, let* $r \in \mathbb{N}$*, and let* $a_0, \dots, a_{r-1} \in A$*,* $b_0, \dots, b_{r-1} \in B$*. Define the partial map* $p : A \to B$ *by* $p(a_i) := b_i$ *for all* $i \in \{0, \dots, r-1\}$*. Then* $p \in \mathrm{Part}(\mathfrak{A}, \mathfrak{B})$ *if and only if for every atomic* S*-formula* $\psi \in L_r^S$ *we have*

$$\mathfrak{A} \vDash \psi[a_0, \dots, a_{r-1}] \text{ if and only if } \mathfrak{B} \vDash \psi[b_0, \dots, b_{r-1}].$$

Proof. First suppose that p is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. Since S is relational, the only terms are variables, and hence the only atomic formulas are equality of variables or relations applied to variables. Suppose first we have $Rv_{i_1} \cdots v_{i_n}$ for $v_{i_1}, \dots, v_{i_n} \in \{v_0, \dots, v_{r-1}\}$ and $R \in S$ is an n-ary relation symbol. Then

$$\begin{aligned}
\mathfrak{A} \vDash Rv_{i_1} \cdots v_{i_n}[a_0, \dots, a_{r-1}] & \text{ if and only if } (a_{i_1}, \dots, a_{i_n}) \in \mathfrak{a}(R) \\
& \text{ if and only if } (p(a_{i_1}), \dots, (a_{i_n})) \in \mathfrak{b}(R) \\
& \text{ if and only if } (b_{i_1}, \dots, b_{i_n}) \in \mathfrak{b}(R) \\
& \text{ if and only if } \mathfrak{B} \vDash Rv_{i_1} \cdots v_{i_n}[b_0, \dots, b_{r-1}].
\end{aligned}$$

Now suppose we have $v_{i_1} \equiv v_{i_2}$ for $v_{i_1}, v_{i_2} \in \{v_0, \dots, v_{r-1}\}$. Then

$$\begin{aligned}
\mathfrak{A} \vDash v_{i_1} \equiv v_{i_2}[a_0, \dots, a_{r-1}] & \text{ if and only if } a_{i_1} = a_{i_2} \\
& \text{ if and only if } p(a_{i_1}) = p(a_{i_2}) \quad (p \text{ is injective}) \\
& \text{ if and only if } b_{i_1} = b_{i_2} \\
& \text{ if and only if } \mathfrak{B} \vDash v_{i_1} \equiv v_{i_2}[b_0, \dots, b_{r-1}].
\end{aligned}$$

Conversely, suppose that for every atomic S-formula $\psi \in L_r^S$ we have $\mathfrak{A} \vDash \psi[a_0, \dots, a_{r-1}]$ if and only if $\mathfrak{B} \vDash \psi[b_0, \dots, b_{r-1}]$. Since S is relational, we need only show that p is injective and that it respects relations symbols. Suppose that $p(a_{i_1}) = p(a_{i_2})$ for some $a_{i_1}, a_{i_2} \in \{a_0, \dots, a_{r-1}\}$. Then $b_{i_1} = b_{i_2}$, so $\mathfrak{B} \vDash v_{i_1} \equiv v_{i_2}[b_0, \dots, b_{r-1}]$. Since this is an atomic formula, by our assumption we get $\mathfrak{A} \vDash v_{i_1} \equiv v_{i_2}[a_0, \dots, a_{r-1}]$, which means that $a_{i_1} = a_{i_2}$. Thus p is injective. Now suppose that $R \in S$ is an n-ary relation symbol and $a_{i_1}, \dots, a_{i_n}$. Then

$$\begin{aligned}
(a_{i_1}, \dots, a_{i_n}) \in \mathfrak{a}(R) & \text{ if and only if } \mathfrak{A} \vDash Rv_{i_1} \cdots v_{i_n}[a_0, \dots, a_{r-1}] \\
& \text{ if and only if } \mathfrak{B} \vDash Rv_{i_1} \cdots v_{i_n}[b_0, \dots, b_{r-1}] \\
& \text{ if and only if } (b_{i_1}, \dots, b_{i_n}) \in \mathfrak{b}(R),
\end{aligned}$$

which shows that p respects relation symbols. Thus p is a partial isomorphism. ∎

Example **194.** Example omitted.

Definition 195. Let S be a first-order language, let $\mathfrak{A} = (A, \mathfrak{a})$ and $\mathfrak{B} = (B, \mathfrak{b})$ be S-structures, and let $(I_n)_{n\in\mathbb{N}}$ be a sequence. We write $(I_n)_{n\in\mathbb{N}} : \mathfrak{A} \cong_f \mathfrak{B}$ if:

for every $n \in \mathbb{N}$, $I_n \neq \varnothing$ and $I_n \subseteq \text{Part}(\mathfrak{A}, \mathfrak{B})$;

Forth Property: for every $n \in \mathbb{N}$ and all $p \in I_{n+1}$, $a \in A$, there exists a $q \in I_n$ such that $p \subseteq q$ and $a \in \text{dom}(q)$;

Back Property: for every $n \in \mathbb{N}$ and all $p \in I_{n+1}$, $b \in B$, there exists a $q \in I_n$ such that $p \subseteq q$ and $b \in \text{range}(q)$.

We say that $\mathfrak{A}$ and $\mathfrak{B}$ are *finitely isomorphic*, and write $\mathfrak{A} \cong_f \mathfrak{B}$, if there is a sequence $(I_n)_{n\in\mathbb{N}}$ such that $(I_n)_{n\in\mathbb{N}} : \mathfrak{A} \cong_f \mathfrak{B}$.

Remark 196. A partial isomorphism in I_n can be extended n times (first to I_{n-1}, then to I_{n-2} ... and finally to I_0).

Definition 197. Let S be a first-order language, let $\mathfrak{A} = (A, \mathfrak{a})$ and $\mathfrak{B} = (B, \mathfrak{b})$ be S-structures, and let $I \subseteq \text{Part}(\mathfrak{A}, \mathfrak{B})$. We write $I : \mathfrak{A} \cong_p \mathfrak{B}$ if $(I)_{n\in\mathbb{N}} : \mathfrak{A} \cong_f \mathfrak{B}$. We say $\mathfrak{A}$ and $\mathfrak{B}$ are *partially isomorphic*, and write $\mathfrak{A} \cong_p \mathfrak{B}$, if there is an I such that $I : \mathfrak{A} \cong_p \mathfrak{B}$.

Remark 198. If two structures are partially isomorphic then they are finitely isomorphic. It can be checked without too much trouble that if $\mathfrak{A} \cong_f \mathfrak{B}$ then $\mathfrak{B} \cong_f \mathfrak{A}$, and similarly for $\cong_p$.

Lemma 199. *Let S be a first-order language and let $\mathfrak{A} = (A, \mathfrak{a})$ and $\mathfrak{B} = (B, \mathfrak{b})$ be S-structures.*

(1) If $\mathfrak{A} \cong \mathfrak{B}$ then $\mathfrak{A} \cong_p \mathfrak{B}$.

(2) If $\mathfrak{A} \cong_p \mathfrak{B}$ then $\mathfrak{A} \cong_f \mathfrak{B}$.

(3) If $\mathfrak{A} \cong_f \mathfrak{B}$ and A or B is finite then $\mathfrak{A} \cong \mathfrak{B}$.

(4) If $\mathfrak{A} \cong_p \mathfrak{B}$ and A and B are countable then $\mathfrak{A} \cong \mathfrak{B}$.

Proof. (1) It is easy to verify that if $\pi : \mathfrak{A} \cong \mathfrak{B}$ then $\{\pi\} : \mathfrak{A} \cong_p \mathfrak{B}$.

(2) It is immediate from definition that if $I : \mathfrak{A} \cong_p \mathfrak{B}$ then $(I)_{n\in\mathbb{N}} : \mathfrak{A} \cong_f \mathfrak{B}$.

(3) We will assume A is finite; the other case is similar. Write $A = \{a_1, \ldots, a_r\}$. Suppose we have $(I_n)_{n\in\mathbb{N}} : \mathfrak{A} \cong_f \mathfrak{B}$. As $I_{r+1} \neq \varnothing$, let $p_0 \in I_{r+1}$. For $i \in \{0, \ldots, r\}$, given $p_i \in I_{r-i+1}$ pick $p_{i+1} \in I_{r-i}$ with $p_i \subseteq p_{i+1}$ and $a \in \text{dom}(p_{i+1})$ (by the Forth Property). Now $p_r \in I_1 \subseteq \text{Part}(\mathfrak{A}, \mathfrak{B})$ with $\text{dom}(p_r) = A$. It remains to show that $\text{range}(p_r) = B$ to

conclude that p_r is an isomorphism. Assume, for a contradiction, that there is some $b \in B \setminus \mathrm{range}(p_r)$. Then there exists a $p \in I_0$ with $b \in \mathrm{range}(p)$ and $p_r \subseteq p$. But this is not possible because there is nothing left in A which could map to b. Contradiction. Thus $p_r : \mathfrak{A} \cong \mathfrak{B}$.

(4) If either A or B is finite, then we can apply (2) and (3) to conlude the desired result. Otherwise, we can write $A = \{a_0, a_1, \ldots\}$ and $B = \{b_0, b_1, \ldots\}$. Suppose we have $I : \mathfrak{A} \cong_p \mathfrak{B}$. As $I \neq \varnothing$, let $p_0 \in I$. For $i \in \{1,3,5,\ldots\}$, given p_{i-1}, pick $p_i \in I$ such that $p_{i-1} \subseteq p_i$ and $a_{\frac{i-1}{2}} \in \mathrm{dom}(p_i)$ (via the Forth Property). For $i \in \{2,4,6,\ldots\}$, given p_{i-1}, pick $p_i \in I$ such that $p_{i-1} \subseteq p_i$ and $b_{\frac{i-2}{2}} \in \mathrm{range}(p_i)$ (via the Back Property). Since they are nested, we can take $p = \bigcup_{n\in\mathbb{N}} p_n$. It is easy to check that $p \in \mathrm{Part}(\mathfrak{A}, \mathfrak{B})$. Moreover, by construction we have $\mathrm{dom}(p) = A$ and $\mathrm{range}(p) = B$. Thus $p : \mathfrak{A} \cong \mathfrak{B}$.

∎

Definition **200.** Define

$$\begin{aligned}\Phi_{\mathrm{dord}} := \{&\forall x \neg < xx,\ \forall x \forall y \forall z((< xy \wedge < yz) \to < xz),\ \forall x \forall y(< xy \vee x \equiv y \vee < yx),\\ &\forall x \forall y(< xy \to \exists z(< xz \wedge < zy)),\ \forall x \exists y < xy,\ \forall x \exists y < yx\}.\end{aligned}$$

A *dense linear ordering without endpoints* is a member of $\mathrm{Mod}^{\{<\}}\,\Phi_{\mathrm{dord}}$.

Theorem **201.** *Any two countable dense linear orderings without endpoints are isomorphic.*

Proof. We will show that any two dense linear orderings without endpoints are partially isomorphic. The desired result then follows by applying the previous lemma.

Suppose $\mathfrak{A}$ and $\mathfrak{B}$ are dense linear orderings without endpoints. Let

$$I = \{p \in \mathrm{Part}(\mathfrak{A}, \mathfrak{B});\ \mathrm{dom}(p) \text{ is finite}\}.$$

We claim that $I : mfA \cong_p \mathfrak{B}$. Note first that $I \neq \varnothing$ since $\varnothing \in I$. And clearly $I \subseteq \mathrm{Part}(\mathfrak{A}, \mathfrak{B})$. Next we check the Forth property. Suppose $p \in I$ and $a \in A$. Write $\mathrm{dom}(p) = \{a_1, \ldots, a_r\}$. Because p is a partial isomorphism and we are dealing with dense linear orders without endpoints, there is some $b \in B$ which has the same relation (with respect to $<$) to $p(a_1), \ldots, p(a_r)$ as a has to $a_1, \ldots, a_r$ (fill in the details). Then one can check $p \cup \{(a,b)\} \in I$. Since $p \subseteq p \cup \{(a,b)\}$ and $a \in \mathrm{dom}(p \cup \{(a,b)\})$, we are done. The Back property is proved analogously. Thus the claim holds. ∎

Definition **202.** Define

$$\Phi_\sigma := \{\forall x(\neg x \equiv 0 \leftrightarrow \exists y \sigma y \equiv x),\ \forall x \forall y(\sigma x \equiv \sigma y \to x \equiv y)\} \cup \bigcup_{m \geq 1} \{\forall x \neg \underbrace{\sigma \cdots \sigma}_{m \text{ times}} x \equiv x\}.$$

A *successor structure* is a member of $\mathrm{Mod}^{\{\sigma,0\}}\,\Phi_\sigma$.

Proposition 203. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be successor structures. Then $\mathfrak{A} \cong_f \mathfrak{B}$.*

Proof. Exercise — see notes from class. ∎

12.2 Fraïssé's Theorem

12.3 Proof of Fraïssé's Theorem

Definition 204. Let S be a first-order language. We define $\mathrm{qr}_S : L^S \to \mathbb{N}$ to be the maximum number of nested quantifiers occurring in a formula:

$$\mathrm{qr}_S(\phi) := 0 \quad \text{for atomic } \phi;$$

$$\mathrm{qr}_S(\neg\phi) := \mathrm{qr}_S(\phi);$$

$$\mathrm{qr}_S((\phi \vee \psi)) := \max\{\mathrm{qr}_S(\phi), \mathrm{qr}_S(\psi)\};$$

$$\mathrm{qr}_S(\exists x\phi) := \mathrm{qr}_S(\phi) + 1.$$

Remark 205. Recall that given a first-order language S, we constructed a relation first-order language S^r. And given an S-structure $\mathfrak{A}$, we constructed an S^r-structure $\mathfrak{A}^r$. One fact we proved was that $\mathfrak{A} \equiv \mathfrak{B}$ if and only if $\mathfrak{A}^r \equiv \mathfrak{B}^r$. If we look back at the construction it is easy to see that

$$\mathrm{Part}(\mathfrak{A}, \mathfrak{B}) = \mathrm{Part}(\mathfrak{A}^r, \mathfrak{B}^r)$$

It follows that $\mathfrak{A} \cong_f \mathfrak{B}$ if and only if $\mathfrak{A}^r \cong_f \mathfrak{B}^r$.

Theorem 206. [Fraïssé's Theorem] *Let S be a finite first-order language and let $\mathfrak{A}$ and $\mathfrak{B}$ be S-structures. Then $\mathfrak{A} \equiv \mathfrak{B}$ if and only if $\mathfrak{A} \cong_f \mathfrak{B}$.*

Proof. Suppose first that $\mathfrak{A} \cong_f \mathfrak{B}$. So there exists a sequence $(I_n)_{n\in\mathbb{N}}$ such that $(I_n)_{n\in\mathbb{N}} : \mathfrak{A} \cong_f \mathfrak{B}$. We claim that for all S-formulas $\phi \in L^S$, for all $r, n \in \mathbb{N}$, all $p \in I_n$, and all $a_0, \ldots, a_{r-1} \in \mathrm{dom}(p)$, if $\mathrm{qr}_S(\phi) \le n$ then

$$\mathfrak{A} \models \phi[a_0, \ldots, a_{r-1}] \text{ if and only if } \mathfrak{B} \models \phi[p(a_0), \ldots, p(a_{r-1})].$$

We proceed by induction on formulas. If ϕ is atomic, then this follows immediately from lemma (193).

The remainder of the proof is omitted for the time being. ∎

Remark 207. We've actually shown that for any $n \in \mathbb{N}$, $\mathfrak{A} \cong_n \mathfrak{B}$ if and only if $\mathfrak{A} \equiv_n \mathfrak{B}$ (see the book for these definitions).

Proposition 208. *Any two dense linear orders without endpoints are elementarily equivalent.*

Proof. We saw in theorem (201) that any two dense linear orders without endpoints are partially isomorphic. We know that this implies that they are finitely isomorphic. Since $\{<\}$ is finite, we can apply Fraïssé's Theorem to conclude that they are elementarily equivalent. ∎

PROPOSITION **209.** *Any two successor structures are elementarily equivalent.*

Proof. We saw in proposition (203) that any two successor structures are finitely isomorphic. Since $\{\sigma, 0\}$ is finite, we can apply Fraïssé's Theorem to conclude that they are elementarily equivalent. ∎

LEMMA **210.** *Let* S *be a first-order language and let* $T \subseteq L_0^S$ *be a theory. Then* T *is complete if and only if any two models of* T *are elementarily equivalent.*

Proof. Suppose T is complete, and let $\mathfrak{A}, \mathfrak{B} \in \mathrm{Mod}^S\, T$ and $\phi \in L_0^S$. If $\phi \in T$, then $\mathfrak{A} \vDash \phi$ and $\mathfrak{B} \vDash \phi$. Otherwise, since T is complete, we know $\neg\phi \in T$. Hence not $\mathfrak{A} \vDash \phi$ and not $\mathfrak{B} \vDash \phi$. Thus $\mathfrak{A} \equiv \mathfrak{B}$.

Conversely, suppose any two models of T are elementarily equivalent. Let $\phi \in L_0^S$. Let $\mathfrak{A}$ be such that $\mathfrak{A} \vDash T$. If $\mathfrak{A} \vDash \phi$, then for any $\mathfrak{B} \vDash T$ we have $\mathfrak{B} \vDash \phi$ (because $\mathfrak{A} \equiv \mathfrak{B}$). Thus $T \vDash \phi$. Because T is a theory this means $\phi \in T$. Otherwise we can do the same thing with $\neg\phi$ (because in this case we know $\mathfrak{A} \vDash \neg\phi$) to conclude $\neg\phi \in T$. Thus T is complete. ∎

COROLLARY **211.** *(1) The theory* $\Phi_{\mathrm{dord}}^{\vDash}$ *of dense linear orderings without endpoints is complete and decidable.*

(2) The theory $\Phi_{\sigma}^{\vDash}$ *of successor structures is complete and decidable.*

Proof. The following proof applies to either claim. By the lemma, to show that the theory is complete, it suffices to prove that any two such structures are elementarily equivalent. But this is exactly the content of the previous propositions. Since axiomatizably complete theories are decidable, the whole result follows (both Φ_{dord} and Φ_{σ} are decidable). ∎

12.4 Ehrenfeucht Games

13 Lindström's Theorem

Introduction to Computability Theory

REMARK **212.** Previously our model of computation had been register machines. We now are going to look at a different model of computation, namely Turing Machines. Note that by Church's Thesis, we will be able to do the exact same things in each of these models.

However, some things are easier to do with Turing Machines, and some things are easier to do with register machines, so there is good reason to prefer one over the other in this section.

Definition 213. A *Turing Machine* consists of a finite program, finitely many states, and has a read-only (oracle) and a read-write (work) infinite tape.

A *Turing program* is a finite list of instructions of the form

$$q_i X Y q_j Z D_1 D_2,$$

where q_i, q_j are states, $X, Y, Z \in \{0, 1\}$, and $D_1, D_2 \in \{L, R\}$.

If a Turing Machine is in state q_i and is reading X on the oracle tape and Y on the work tape, and if $q_i X Y q_j Z D_1 D_2$ is an instruction in the program, then the machine will write Z on the work tape in place of Y, shift its work tape reading head one place to the left/right (depending on D_1), and shift its work tape reading head one place to the left/right (depending on D_2). The state q_0 is the halting state.

We can effectively list all Turing programs. Let $P_0, P_1, P_2, \ldots$ be such a list. To each program P_i we can associate a partial function $\phi_i : \mathbb{N} \to \mathbb{N}$ as follows. If P_i, started in state q_1 with $n + 1$ consecutive 1's on the work tape and all 0's on the oracle tape, with work reading head at the leftmost 1, eventually reaches the halting state q_0, then we write $\phi_i(n) \downarrow$, and we let $\phi_i(n)$ be the number of 1's on the work tape when it halts. If P_i, when started as above, never halts, we write $\phi_i(n) \uparrow$. We let $W_i := \mathrm{dom}(\phi_i)$.

Definition 214. Let $A \subseteq \mathbb{N}$. We say A is *computable* if there is an $i \in \mathbb{N}$ such that $\chi_A = \phi_i$.

Definition 215. Let $A \subseteq \mathbb{N}$. We say A is *computably enumerable* if there is an $i \in \mathbb{N}$ such that $\mathrm{dom}(\phi_i) = A$.

Remark 216. These notions correspond to decidable and enumerable respectively.

Also note that $W_0, W_1, W_2, \ldots$ is an effective listing of all computably enumerable sets.

Definition 217. Let $f : \mathbb{N} \to \mathbb{N}$ be a partial function. We say f is *partially computable* if there is an $i \in \mathbb{N}$ such that $f = \phi_i$.

Definition 218. Let $f : \mathbb{N} \to \mathbb{N}$ be a function. We say f is *computable* (or *total*) if there is an $i \in \mathbb{N}$ such that $f = \phi_i$.

Definition 219. Let $e, s, x, y \in \mathbb{N}$. We write $\phi_{e,s}(x) \downarrow$ and $\phi_{e,s}(x) := y$ if program P_e, started with input x and empty oracle, halts within s steps and outputs y. We write

$\phi_{e,s}(x)\uparrow$ if program P_e, started with input x and empty oracle, has not yet halted within s steps. We let

$$W_{e,s} := \{y \in \mathbb{N};\ \phi_{e,s}(y)\downarrow\}.$$

Definition 220. We define $\langle \cdot,\cdot \rangle : \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ by

$$\langle x,y\rangle := \frac{1}{2}\left(x^2+2xy+y^2+3x+y\right)$$

for all $(x,y)\in\mathbb{N}\times\mathbb{N}$.

Remark 221. The standard pairing function $\langle\cdot,\cdot\rangle$ is a bijection, and can be computed (say, using a Turing Machine).

Definition 222. Let $R\subseteq\mathbb{N}^2$ be a binary relation. We say R is *computable* if

$$\{\langle x,y\rangle\,;\,(x,y)\in R\}$$

is computable.

Definition 223. Let $R\subseteq\mathbb{N}^2$ be a binary relation. We say R is *computably enumerable* if

$$\{\langle x,y\rangle\,;\,(x,y)\in R\}$$

is computably enumerable.

Definition 224. Let $A\subseteq\mathbb{N}$. We say A is Σ_1 if there is a computable relation $R\subseteq\mathbb{N}^2$ such that for all $x\in\mathbb{N}$, $x\in A$ if and only if there is a $y\in\mathbb{N}$ such that $(x,y)\in R$.

Theorem 225. *Let $A\subseteq\mathbb{N}$. Then A is computably enumerable if and only if A is Σ_1.*

Proof. First suppose that A is computably enumerable. Then $A=W_e$ for some $e\in\mathbb{N}$. Note $x\in W_e$ if and only if there is an $s\in\mathbb{N}$ such that $x\in W_{e,s}$. Moreover, the relation $\{(x,s)\in\mathbb{N}^2; x\in W_{e,s}\}$ is computable. Indeed, given $n\in\mathbb{N}$, we can compute $x,s\in\mathbb{N}$ such that $n=\langle x,s\rangle$. From here we can run P_e with input x for s steps and see if it has halted yet. Thus A is Σ_1.

Conversely, suppose A is Σ_1. So there is a computable relation $R\subseteq\mathbb{N}^2$ such that, $x\in A$ if and only if there exists a $y\in\mathbb{N}$ such that $(x,y)\in R$. Consider the program that on input $x\in\mathbb{N}$, asks for each $y\in\mathbb{N}$ in turn (in order) whether $(x,y)\in R$, and halts if there is ever an affirmative answer. Then since R is computable, $P=P_e$ for some $e\in\mathbb{N}$, and $A=\mathrm{dom}(\phi_e)=W_e$. Thus A is computably enumerable. ∎

Theorem 226. *Let $A \subseteq \mathbb{N}$ be non-empty. Then A is computably enumerable if and only if it is the range of a computable function.*

Proof. First suppose that $f : \mathbb{N} \to \mathbb{N}$ is computable and $A = \text{range}(f)$. Note $x \in A$ if and only if there exists a $z \in \mathbb{N}$ such that $x = f(z)$. Since f is computable, the relation $\{(x, z); x = f(z)\}$ is computable. Hence A is Σ_1. But we just saw that this means A is computably enumerable.

Conversely, suppose that A is computably enumerable. As A is non-empty, let $a \in A$. Define $f : \mathbb{N} \to \mathbb{N}$ by

$$f(\langle x, s\rangle) := \begin{cases} x & \text{if } x \in W_{e,s}, \\ a & \text{otherwise.} \end{cases}$$

Then f is computable and $\text{range}(f) = A$. ∎

Theorem 227. *There is no effective listing of the computable functions.*

Proof. Assume, for a contradiction, that $f_0, f_1, f_2, \ldots$ is an effective listing of the computable functions. Define $g : \mathbb{N} \to \mathbb{N}$ by $g(n) := f_n(n) + 1$ for all $n \in \mathbb{N}$. As each f_n is computable, g is computable. But $g(n) \neq f_n(n)$ and so $g \neq f_n$, for each $n \in \mathbb{N}$, so g is not in our list. Contradiction. ∎

Definition 228. We define

$$K := \{e \in \mathbb{N}; \phi_e(e)\downarrow\}$$

to be the usual halting set, and

$$K_0 := \{\langle e, n\rangle ; (e, n) \in \mathbb{N}^2, \phi_e(n)\downarrow\}.$$

Proposition 229. *The set K is not computable.*

Proof. Assume, for a contradiction, that K is computable. Then the function

$$g(x) := \begin{cases} \phi_x(x) + 1 & \text{if } x \in K, \\ 0 & \text{otherwise,} \end{cases}$$

is computable. So $g = \phi_e$ for some $e \in \mathbb{N}$. As g is total, $g(e)\downarrow$. Hence $\phi_e(e)\downarrow$, and so $e \in K$. But now $\phi_e(e) = g(e) = \phi_e(e) + 1$, which is absurd. ∎

Proposition 230. *The set K_0 is not computable.*

Proof. Note $e \in K$ if and only if $\langle e, e\rangle \in K_0$. So if K_0 were computable, then K would be computable, which it isn't. Thus K_0 is not computable. ∎

DEFINITION **231.** Let $A, B \subseteq \mathbb{N}$. We say A is *many-one reducible to* B, and write $A \leq_m B$, if there is a computable function f such that $x \in A$ if and only if $f(x) \in B$.

REMARK **232.** We have seen $K \leq_m K_0$ (via $n \mapsto \langle n, n \rangle$).

THEOREM **233.** *Let* $A, B \subseteq \mathbb{N}$.

(a) If $A \leq_m B$ *and* B *is computable then* A *is computable.*

(b) If $A \leq_m B$ *and* B *is computably enumerable then* A *is computably enumerable.*

Proof. (a) Let $x \in \mathbb{N}$. To compute whether $x \in A$, first compute $f(x)$, then compute whether $f(x) \in B$.

(b) As B is computably enumerable, there exists an $e \in \mathbb{N}$ such that $B = W_e$. Note $x \in A$ if and only if $f(x) \in B$, if and only if there is an $s \in \mathbb{N}$ such that $f(x) \in W_{e,s}$. This means A is Σ_1, and is hence computably enumerable.

∎

THEOREM **234.** [s-m-n THEOREM] *If* ϕ *is a partially computable function on* $\mathbb{N}^2$ *(corresponding to starting a Turing Maching with* $x+1$*-many* 1*'s, followed by a* 0*, followed by* $y+1$*-many* 1*'s), then there exists a computable function* f *such that*

$$\phi_{f(x)}(y) = \psi(x, y)$$

for all $(x, y) \in \mathbb{N}^2$.

Proof. Omitted. ∎

REMARK **235.** We have seen that $K \leq_m K_0$. We now show that $K_0 \leq_m K$, so that the two sets are equivalent in the sense of many-one reducibility.

PROPOSITION **236.** *We have* $K_0 \leq_m K$.

Proof. Define a partially computable function ψ by

$$\psi(\langle e, n \rangle, y) := \begin{cases} 0 & \text{if } \phi_e(n) \downarrow, \\ \uparrow & \text{otherwise.} \end{cases}$$

The program for ψ on input (x, y) first decodes x into $\langle e, n \rangle$, and then runs P_e with input n. If P_e halts, it outputs 0.

By the s-m-n theorem, there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that

$$\phi_{f(\langle e, n \rangle)}(y) = \psi(\langle e, n \rangle, y).$$

We claim that $x \in K_0$ if and only if $f(x) \in K$. First suppose $\langle e, n \rangle \in K_0$. Then $\phi_e(n)\downarrow$, so $\psi(\langle e,n\rangle, f(\langle e,n\rangle)) = 0$ by definition. This means $\phi_{f(\langle e,n\rangle)}(f(\langle e,n\rangle)) = 0$, and so in particular $\phi_{f(\langle e,n\rangle)}(f(\langle e,n\rangle))\downarrow$. Thus $f(\langle e,n\rangle) \in K$. Conversely, suppose $\langle e,n\rangle \notin K_0$. Then $\phi_e(n)\uparrow$, so $\psi(\langle e,n\rangle, f(\langle e,n\rangle))\uparrow$ by definition. This means $\phi_{f(\langle e,n\rangle)}(f(\langle e,n\rangle))\uparrow$. Thus $f(\langle e,n\rangle) \notin K$. Thus the claim holds and so $K_0 \leq_m K$. ∎

Remark **237.** Suppose W_e is a computably enumerable set, for some $e \in \mathbb{N}$. Then $n \in W_e$ if and only if $\langle e,n\rangle \in K_0$. Hence $W_e \leq_m K_0$. It is easy to check that if $A \leq_m B$ and $B \leq_m C$ then $A \leq_m C$, for any $A, B, C \subseteq \mathbb{N}$. Thus we conclude that $W_e \leq_m K$, for any $e \in \mathbb{N}$.

We call K (and K_0) *complete* since they can uniformly compute any computably enumerable set. That is to say, there exists a computable function $F : \mathbb{N}^2 \to \mathbb{N}$ such that for all $e \in \mathbb{N}$, $y \in W_e$ if and only if $F(e,y) \in K$.

We would hope that this relation $\leq_m$ captures the complexity (computability-wise) of a set. On the other hand, it also makes sense that for any set $A \subseteq \mathbb{N}$, A and $\bar{A}$ would be equivalent in such a sense. However, if A is any non-computable, computably enumerable set, then $\bar{A} \not\leq_m A$.

Definition **238.** Let $A, B \subseteq \mathbb{N}$. We say A is *Turing reducible to* B, and write $A \leq_T B$, if there is a Turing program P_e such that if B is on the oracle tape, then P_e started on input n, after finitely many steps halts with 1 on the work tape if $n \in A$ and 0 on the work tape if $n \notin A$.

Definition **239.** Let $B \subseteq \mathbb{N}$ and let $e \in \mathbb{N}$. Just as the program P_e induced a partial function ϕ_e by putting all zeros on the oracle tape, the program P_e and the set B induce a partial function Φ_e^B, by putting B on the oracle tape. We let $W_e^B := \mathrm{dom}(\Phi_e^B)$.

For $x, s, y \in \mathbb{N}$, we write $\Phi_{e,s}^B(x) := y$ if P_e started with input x and oracle B halts within s steps, having output y. For $x, y \in \mathbb{N}$, we write $\Phi_e^B(x)\downarrow$ and $\Phi_e^B(x) := y$ if there exists an $s \in \mathbb{N}$ such that $\Phi_{e,s}^B(x) = y$. For $x \in \mathbb{N}$, we write $\Phi_e^B(x)\uparrow$ if there is no $(y,s) \in \mathbb{N}^2$ such that $\Phi_{e,s}^B(x) = y$.

Remark **240.** If $A \leq_T B$ we will write $A = \Phi_e^B$, even though the former is a set and the latter is a partial function.

Also note that $\Phi_e^{\varnothing} = \phi_e$.

Moreover, if $\Phi_e^B(x)\downarrow$, then it halts after finitely many steps, and so it may only have looked at a finite segment of the oracle B. That is to say, if $\Phi_e^B(x) = y$ then there is a finite initial segment $\sigma \subseteq B$ such that all queries to B are within the segment σ. In this case we write $\Phi_e^\sigma(x) := y$. Note that if $\Phi_{e,s}^\sigma(x) = y$ then for any $\tau \supseteq \sigma$ and any $t \geq s$ we have $\Phi_{e,t}^\tau(x) = y$.

Example 241. Consider the program P_e which on input $n \in \mathbb{N}$, sums the numbers on the oracle tape between $2n+1$ and $3n$ (inclusive). For any $n \in \mathbb{N}$, we query a finite initial segment of the oracle tape, but for any $m \in \mathbb{N}$ we can find an $n \in \mathbb{N}$ such that P_e on input n queries position m of the oracle.

Note $\Phi_e^{\varnothing}(n) = 0$ and $\Phi_e^{\mathbb{N}}(n) = n$.

Definition 242. Let $A, B \subseteq \mathbb{N}$. We say A is *computably enumerable in* B if there is a program P_e such that $A = W_e^B$.

Definition 243. Let $A, B \subseteq \mathbb{N}$. We say A is Σ_1 *in* B if there is a B-computable binary relation R such that $x \in A$ if and only if there exists a $y \in \mathbb{N}$ such that $(x, y) \in R$.

Theorem 244. *Let* $A, B \subseteq \mathbb{N}$*. Then* A *is computably enumerable in* B *if and only if* A *is* Σ_1 *in* B*.*

Proof. Suppose first that A is computable enumerable in B. So there is an $e \in \mathbb{N}$ such that $A = W_e^B$. Consider the relation $R = \{(x, s) \in \mathbb{N}^2; x \in W_{e,s}^B\}$. This is B-computable, clearly. And $x \in A$ if and only if there exists an $s \in \mathbb{N}$ such that $(x, s) \in R$. Thus A is Σ_1 in B.

Conversely, suppose that A is Σ_1 in B. So there exists a B-computable relation R such that $x \in A$ if and only if there exists a $y \in \mathbb{N}$ such that $(x, y) \in R$. Consider the program P_e which on input $n \in \mathbb{N}$, checks for each $y \in \mathbb{N}$ one at a time if $(n, y) \in R$ (since this is B-computable). Indeed, this will halt on input x with oracle B if and only if there is a $y \in \mathbb{N}$ such that $(x, y) \in R$, if and only if $x \in A$. Thus $A = W_e^B$ is computably enumerable in B. ∎

Definition 245. Let $A \subseteq \mathbb{N}$. We define the *jump* of A to be

$$A' := \{e \in \mathbb{N}; \Phi_e^A(e)\downarrow\}.$$

Example 246. We have $\varnothing' = K$, the halting set.

Definition 247. Let $A, B \subseteq \mathbb{N}$. We write $A \equiv_T B$ if $A \leq_T B$ and $B \leq_T A$. We write $A <_T B$ if $A \leq_T B$ but $A \not\equiv_T B$.

Proposition 248. *Let* $A, B \subseteq \mathbb{N}$*.*

(1) The jump A' *is computably enumerable in* A*.*

(2) We have $A <_T A'$*.*

(3) If A *is computably enumerable in* B *then* $A \leq_T B'$*.*

(4) If $A \leq_T B$ then $A' \leq_T B'$.

Proof. Exercise. ∎

Remark 249. If $\varnothing \leq_T A \leq_T \varnothing'$ then $\varnothing' \leq_T A' \leq_T \varnothing''$.

Definition 250. Let $A \subseteq \mathbb{N}$. We say A is *low* if $A' = \varnothing'$. We say A is *high* if $A \leq_T \varnothing'$ and $A' \equiv_T \varnothing''$.

Theorem 251. *There exists a non-computable low set.*

Proof. We will build such a set A by stages. At each stage $s+1$ we will have $\alpha_s \subseteq \alpha_{s+1}$ be finite binary strings. We will then let $A = \bigcup_{s \in \mathbb{N}} \alpha_s$.

Our construction will not be computable, but it will be $\varnothing'$-computable. That is, at each stage $s+1$, we will ask finitely many questions of a $\varnothing'$-oracle. The set A will then be $\varnothing'$-computable, since to compute whether $x \in A$, run the construction until $|\alpha_s| > x$, at which point $x \in A$ if and only if $\alpha_s(x) = 1$.

As we build A, we will meet for each $e \in \mathbb{N}$ the requirement

$$R_e : \quad \chi_A \neq \phi_e.$$

This will ensure that A is not computable. We will accomplish these tasks during the even stages of the construction. During the odd stages of the construction, we will ensure that A is low. To do this we will decide whether or not $\Phi_e^A(e)\downarrow$. Since the construction will be $\varnothing'$-computable, this will ensure that $A' \leq_T \varnothing'$.

We begin the construction. Let $\alpha_0 := \varnothing$.

Suppose we are at stage $2e+1$, given α_{2e}. Ask the $\varnothing'$-oracle the question

$$\exists \sigma \supseteq \alpha_{2e} \exists t \Phi_{e,t}^{\sigma}(e)\downarrow.$$

(Indeed, this is a Σ_1-question, so given α_{2e} we can effectively find the appropriate location of the $\varnothing'$-oracle to check, where a 1 means 'yes' and a 0 means 'no'.) If the answer is 'yes', then we can effectively find such a σ, and set $\alpha_{2e+1} := \sigma$. If the answer is 'no', then we let $\alpha_{2e+1} := \alpha_{2e}$.

Suppose we are at stage $2e+2$, given α_{2e+1}. Ask the $\varnothing'$-oracle the question

$$\exists t \phi_{e,t}(|\alpha_{2e+1}|)\downarrow.$$

If the answer is 'yes', then compute $\phi_e(|\alpha_{2e+1}|)$. If $\phi_e(|\alpha_{2e+1}|) = 0$ then let $\alpha_{2e+2} := \alpha_{2e+1}1$, and otherwise let $\alpha_{2e+2} := \alpha_{2e+1}0$. If the answer is 'no', then let $\alpha_{2e+2} := \alpha_{2e+1}0$.

This completes the construction. Note that at each even step we strictly increase the length of α_s, so we can let $A = \bigcup_{s \in \mathbb{N}} \alpha_s$.

First we verify that A is low. We will describe a $\varnothing'$-computable procedure to decide for each $e \in \mathbb{N}$ whether $e \in A'$. First we use the $\varnothing'$-oracle to compute α_{2e}. Next ask the question

$$\exists \sigma \supseteq \alpha_{2e} \exists t \Phi^{\sigma}_{e,t}(e) \downarrow .$$

If the answer is 'yes', then we picked α_{2e+1} such that $\Phi_e^{\alpha_{2e+1}}(e) \downarrow$, and hence $\Phi_e^A(e) \downarrow$ and $e \in A'$. If the answer is 'no', then we know $\Phi_e^A(e) \uparrow$ (if it didn't, then the answer to the question would be 'yes'), and so $e \notin A'$. Thus A is low.

Finally we verify that A is not computable. We need to show $\chi_A \neq \phi_e$ for each $e \in \mathbb{N}$. Suppose ϕ_e is total (ie computable) (if it isn't then clearly $\phi_e \neq \chi_A$). Then at step $2e+2$, the answer to the question

$$\exists t \phi_{e,t}(|\alpha_{2e+1}|) \downarrow$$

was yes. But this means we defined α_{2e+2} so that $\alpha_{2e+2}(|\alpha_{2e+1}|) \neq \phi_e(|\alpha_{2e+1}|)$. Hence $\chi_A \neq \phi_e$. Thus A is not computable. ∎

LEMMA **252.** [LIMIT LEMMA] *Let $g : \mathbb{N} \to \mathbb{N}$ be a function. Then g is $\varnothing'$-computable if and only if there exists a computable function $f : \mathbb{N}^2 \to \mathbb{N}$ such that for all $x \in \mathbb{N}$, $g(x) = \lim_s f(x, s)$.*

Proof. Suppose first that such an f exists. We can $\varnothing'$-compute g as follows. Let $x \in \mathbb{N}$. For each $s \in \mathbb{N}$, ask

$$\exists t > s f(x,t) \neq f(x,s).$$

If the answer is 'no' at some stage s, then $g(x) = \lim_t f(x,t) = f(x,s)$. If the answer is 'yes' at some stage s, then continue to the next stage $s+1$. Since g is total and $g(x) = \lim_s f(x,s)$, this procedure must eventually terminate, and produce the correct answer.

Conversely, suppose g is $\varnothing'$-computable. Then $g = \Phi_e^K$ for some $e \in \mathbb{N}$. Let $(K_s)_{s\in\mathbb{N}}$ be a computable enumeration of $K = \varnothing'$. Define

$$f(x,s) := \begin{cases} \Phi^{K_s}_{e,s}(x) & \text{if it converges,} \\ 0 & \text{otherwise.} \end{cases}$$

For any $x \in \mathbb{N}$, since $g(x) = \Phi_e^K(x)$, there is some $\sigma \subset K$ and some $t \in \mathbb{N}$ such that $g(x) = \Phi^{\sigma}_{e,t}(x)$. Let $s_0 \geq t$ be such that for all $n \leq |\sigma|$, $n \in K$ if and only if $n \in K_{s_0}$. Then for all $s \geq s_0$ we have $\Phi^{K_s}_{e,s}(x) = g(x)$. Thus $g(x) = \lim_s f(x,s)$. ∎

THEOREM **253.** *There exists a non-computable, low, computably enumerable set.*

Proof. We will construct such a set A by stages, similarly to before. We will meet the requirements

$$\begin{aligned} P_e &: \quad \chi_A \neq \phi_e, \\ N_e &: \quad \exists^{\infty} s \Phi^{A_s}_{e,s}(e) \downarrow \implies \Phi_e^A(e) \downarrow , \end{aligned}$$

for each $e \in \mathbb{N}$, where A_s is the finite amount of A enumerated by the end of stage s.

The requirements P_e clearly ensure that S is not computable (just like last time). The requirements N_e will ensure that A is low. Indeed, suppose A meets the N_e requirements. Let

$$f(e,s) := \begin{cases} 1 & \text{if } \Phi_{e,s}^{A_s}(e)\downarrow, \\ 0 & \text{otherwise.} \end{cases}$$

Because N_e holds, we have $\lim_s f(e,s) = \chi_{A'}(e)$ (check). Therefore A' is limit computable and hence $\varnothing'$-computable (that is, A is low).

Plan to meet requirement P_e: Choose some $x_e \in \mathbb{N}$. Wait for a stage s where $\phi_{e,s}(x_e)\downarrow= 0$. If this never occurs, then keep x_e out of A (and so $\chi_A(x_e) = 0 \neq \phi_e(x_e)$). If such a stage s is encountered, then enumerate x_e into A (and so $\chi_A(x_e) = 1 \neq 0 = \phi_e(x_e)$).

Plan to meet requirement N_e: It would be nice not to enumerate into A, however this doesn't match the strategy for P_e's. If we see that $\Phi_{e,s}^{A_s}(e)\downarrow$, then we will attempt to preserve the "use" of this computation. That is, we will try not to enumerate into A below $|\sigma|$, where σ is the least such that $\Phi_{e,s}^{\sigma}(e)\downarrow$.

We will make use of witnesses $x_{e,s}$, where we will ensure that $x_e = \lim_s x_{e,s}$ exists and that $\chi_A(x_e) \neq \phi_e(x_e)$. If $\Phi_{e,s}^{A_s}(e)\downarrow$ then there is some finite $\sigma \subseteq A_s$ such that $\Phi_{e,s}^{\sigma}(e)\downarrow$. So if we do not enumerate any number less than $|\sigma|$ into A after stage s, then $\Phi_e^A(e)\downarrow$. To attempt to preserve computations, we will have restraint functions $r(e,s)$. The idea is that only requirements P_i for $i \le e$ will be allowed to have witnesses below the restraint for requirement N_e.

We begin the construction. Let $r(e,0) = 0$ and $x_{e,0} = \langle e, 0\rangle$ for each e, and let $A_0 = \varnothing$.

Suppose we are at stage $s+1$. First, suppose $\phi_{e,s+1}(x_{e,s})\downarrow= 0$ for some unsatisfied P_e with $e \le s$. Then enumerate $x_{e,s}$ into A_{s+1} and declare P_e satisfied.

Next, for all $e \le s$ such that $\Phi_{e,s+1}^{A_{s+1}}(e)\downarrow$, let $r(e,s+1)$ be the "use of the computation" (ie the greatest number of the oracle queried during the computation, plus one).

Finally, for all $e \le s$, if P_e is not satisfied then let $x_{e,s+1}$ be the least number y of the form $\langle e, z\rangle$ such that $y > r(i,s+1)$ for all $i < e$. Otherwise let $x_{e,s+1} = x_{e,s}$. This completes the construction. The verification that the construction produces the desired set A is the content of the following lemmas. ∎

Lemma 254. *For each $e \in \mathbb{N}$, there is at most one stage s when $x_{e,s}$ is enumerated into A.*

Proof. If $x_{e,s}$ is enumerated into A at stage s, then P_e is declared satisfied, and there is no more enumeration of witnesses of the form $x_{e,t}$ into A. ∎

Lemma 255. *For all $e \in \mathbb{N}$, $\lim_s r(e,s)$ exists (and is finite).*

Proof. Suppose $r(e,s) \neq 0$. Then $\Phi_{e,s}^{A_s \upharpoonright r(e,s)}(e)\downarrow$. Then $r(e,s+1) = r(e,s)$ unless $A_{s+1} \upharpoonright r(e,s) \neq A_s \upharpoonright r(e,s)$ (ie unless some number less than $r(e,s)$ is enumerated into A at stage

$s+1$). But by our construction, this number would be of the form $x_{i,s}$ with $i \le e$. So by the previous lemma, this can only happen finitely often. Hence the sequence eventually becomes constant. ∎

LEMMA **256.** *The requirements* N_e *are met.*

Proof. Let $e \in \mathbb{N}$. Let s be such that $r(e,t) = r(e,s)$ for all $t \ge s$ (this is possible by the previous lemma). If $r(e,s) = 0$, then $\Phi_{e,t}^{A_t} \uparrow$ for all $t > s$, and so there is nothing to meet. If $r(e,s) = 0$, then $\Phi_{e,t}^{A_t} \downarrow = \Phi_{e,s}^{A_s}(e)$ for all $t > s$, so $\Phi_e^A(e) \downarrow$, meeting the requirement N_e. ∎

LEMMA **257.** *The requirements* P_e *are met.*

Proof. Let $e \in \mathbb{N}$. Let s be such that $r(i,s) = \lim_t r(i,t)$ for all $i < e$. Then $x_{e,t} = x_{e,s}$ for all $t \ge s$. If $\phi_{e,t}(x_{e,t}) \downarrow = 0$, then $x_{e,t} \in A$ so $\chi_A \ne \phi_e$. If $\phi_e(x_{e,s}) \ne 0$, then $x_{e,s} \notin A$, so $\chi_A \ne \phi_e$. ∎