Mathematics 114L | Spring 2013 | D.A. Martin

Mathematical Logic

1 First-Order Languages.

Symbols. All first-order languages we consider will have the following symbols:

(i)	variables	$v_1, v_2, v_3, \ldots;$
(ii)	connectives	$\neg, \rightarrow;$
(iii)	parentheses	$(\, , \,)\,;$
(iv)	identity symbol	$=;$
(v)	quantifier	$\forall\,.$

For each $n \geq 0$, a language might have:

(vi) one or more n-place predicate symbols;
(vii) one or more n-place function symbols.

We call 0-place predicate symbols *sentence symbols*, and we call 0-place function symbols *constants*.

Remark. We don't worry about what can count as a symbol, but it is important that in a single language nothing can be a symbol of two different kinds. For example, F cannot be simultaneously a function symbol and a predicate symbol.

A language is determined by its predicate and function symbols, so will think of a language as the set of its predicate and function symbols.

Examples:

(1) The language of identity: $\emptyset$.

(2) The language of ordering: (or one of them) $\{\leq\}$.

(3) The language of arithmetic: $\{0, S, +, \cdot, \leq\}$.

Here $\emptyset$ is the empty set, $\leq$ is a two-place predicate symbol, 0 is a constant, S is a one-place function symbol, and $+$ and $\cdot$ are two-place function symbols. Often $\leq$ is omitted from the language of arithmetic.

For the rest of this section, fix a language $\mathcal{L}$.

Terms of $\mathcal{L}$:

(1) Each variable or constant is a term.

(2) If $n \geq 1$, if f is an n-place function symbol, and if $t_1, \ldots, t_n$ are terms, then $ft_1 \ldots t_n$ is a term.

(3) Nothing is a term unless its being one follows from (1)–(2).

We will often write, e.g., "$f(t_1, t_2)$" for "ft_1t_2" for ease of readability.

Formulas of $\mathcal{L}$:

(i) If t_1 and t_2 are terms, then $t_1 = t_2$ is a formula.

(ii) If $n \geq 0$, if P is an n-place predicate symbol, and if $t_1, \ldots, t_n$ are terms, then $Pt_1 \ldots t_n$ is a formula.

(iii) If φ is a formula, then so is $\neg\varphi$.

(iv) If φ and ψ are formulas, then so is $(\varphi \to \psi)$.

(v) If φ is a formula and x is a variable, then $\forall x\varphi$ is a formula.

(vi) Nothing is a formula unless its being one follows from (i)–(iii).

Formulas given by (i) or (ii) are called *atomic* formulas. Another way to state the definition of "formula" is to say that the collection of all formulas is gotten by starting with the atomic formulas and closing under the operations $\varphi \mapsto \neg\varphi$ and $(\varphi, \psi) \mapsto (\varphi \to \psi)$. Similarly the collection of all terms is gotten by starting with the *atomic* terms (the constants and the variables) and closing under the operation given by clause (ii) of the definition of term.

We think of terms and formulas as finite sequences of symbols. Thus all terms and formulas have a *length*. For example, if f is a two-place function symbol then the length $\ell\text{h}(fv_1v_2)$ of the term fv_1v_2 is 3.

Remark. To avoid confusion between symbols and finite sequences of them, we need to require that no finite sequence of symbols can be a symbol. Thus the variable v_1 should be distinguished from the sequence of length one consisting of v_1. We will frequently violate this requirement. Indeed, we have already done so in declaring that each variable or constant is a term. Later in the course, when we introduce Gödel numbers, we will have to start paying attention to the requirement.

If we want to prove that all formulas or all terms have some property P, a good method to employ is *proof by induction on length*. To prove by induction on length that all formulas have property P, one must demonstrate the the following fact:

(†) For every formula φ, if every formula shorter than φ has property P then φ has P.

(There is an analogous statement for the case of terms, and we also call it (†).) To see that proving (†) does indeed prove that all formulas have P, assume that (†) is true but that not all formulas have P. There must be a number n that is the shortest length of any formula that lacks P. Let φ be a formula of length n that lacks P. Every formula shorter than φ has P, and this contradicts (†).

An important fact about terms and formulas is that they are *syntactically unambiguous*. Consider the case of formulas. Suppose there were a formula φ which was both $(\psi \to \chi)$ and $(\psi' \to \chi')$ but that ψ was not the same formula as ψ'. Then φ would be syntactically ambiguous: it would come in two different ways by clause (iv) in the definition of formula. Why does this kind of ambiguity seem possible? A long conditional formula $(\psi \to \chi)$ can contain many occurrences of the symbol $\to$. Perhaps, e.g., one of the occurrences of $\to$ that occurs before the end of the formula ψ could also function as the central $\to$ of the formula.

Another word for syntactic unambiguity is *unique readability*, and that is the word we will mainly use.. To prove unique readability for terms, we first prove the following fact.

Lemma 1.1. *No proper initial segment of a term is a term.*

Proof. We prove by induction on length that every term has the property P of being a term with no proper initial segment that is a term. To do this we must prove the terms version of (†). Assume, then, that t is a term and that every term shorter than t has property (ii). We must prove that t has P. By clause (iii) in the definition of term, t must either be an atomic term or a term of the form $ft_1 \dots t_n$. This means we have two cases to consider.

Case 1. t is a constant or variable. Since constants or variables have length 1, the only proper initial segment of t is the empty sequence, which is obviously not a term.

Case2. For some $n \geq 1$, t is $ft_1 \dots t_n$ for some n-place function symbol f and some terms $t_1, \dots t_n$. Suppose that t' is a proper initial segment of t that is a term. Then t' must be $ft'_1, \dots t'_n$ for some terms $t'_1, \dots t'_n$. Notice that all of $t_1, \dots, t_n, t'_1, \dots t'_n$ are shorter terms than t. Hence all of them have P, i.e., none of them have proper initial segments that are terms. Hence neither t_1 nor t'_1 is a proper initial segment of the other, and so t_1 is the same as t'_1. Repeating this reasoning $n-1$ more times, we deduce that t_i and t'_i are the same for $1 \leq i \leq n$. But then $t = t'$, contradicting the fact that t' is a proper initial segment of t. □

Theorem 1.2 (Unique Readability for Terms). *Every non-atomic term is $ft_1 \dots t_n$ for a unique $n \geq 1$, a unique n-place function symbol f, and unique terms $t_1, \dots, t_n$.*

Proof. We need only prove uniqueness. Assume that t is both $ft_1 \dots t_n$ and $f't'_1 \dots t'_n$. Clearly f and f' are the same. Using Lemma 1.1, we get successively that t_1 is $t'_1, \dots, t_n$ is t'_n. □

Lemma 1.3. *No proper initial segment of a formula is a formula.*

Proof. We use induction on length, with P the property of being a formula with no proper initial segment that is a formula. To prove (†), assume that φ is a formula and that every formula shorter than φ has P. There are five cases we must deal with, corresponding clauses (i)-(v) in the definition of formula.

Case 1. φ is $t_1 = t_2$ for some terms $t_1 = t_2$. It is easy to show that terms contain no symbols other that constants, variables, and function symbols, and thus any proper initial segment of φ that is a formula has to be $t_1 = t'$, for t' a term that is a proper initial segment of t_2. By Lemma 1.1, there is no such term.

Case 2. φ is $Qt_1 \dots t_n$ for some n, some n-place predicate symbol Q, and some terms $t_1, \dots, t_n$. Any proper initial segment of φ that is a formula would have to be $Qt'_1 \dots t'_n$ for some terms $t'_1 \dots t'_n$. An argument like that for Case 2 of the proof of Lemma 1.1 show that this is impossible.

Cases 3, 4, and 5 are Exercise 1.1. □

Exercise 1.1. Supply cases 3, 4, and 5 of the proof of Lemma 1.3.

Theorem 1.4 (Unique Readability for Formulas). *For any formula φ, exactly one of the following holds.*

(1) *There are unique terms t_1 and t_2 such that φ is $t_1 = t_2$.*

(2) *There are unique $n \geq 0$, Q, and $t_1, \ldots, t_n$ such that Q is an n-place predicate symbol, $t_1, \ldots, t_n$ are terms, ad φ is $Qt_1 \ldots t_n$.*

(3) *There is a unique formula ψ such that φ is $\neg\psi$.*

(4) *There are unique formulas ψ and χ such that φ is $(\psi \to \chi)$.*

(5) *There are a unique formula ψ and a unique variable x such that φ is $\forall x\psi$.*

Proof. At most one of the conditions holds. This is because formulas satisfying (1) begin with constants, variables or function symbols, those satisfying (2) begin with predicate symbols, those satisfying (3) begin with $\neg$; those satisfying (4) begin with (, and those satisfying (5) begin with $\forall$.

By the definition of formula, all we need to prove is uniqueness,for each of the five kinds of formulas. The only non-trivial case of uniqueness is that for Case 4, formulas φ of the form $(\psi \to \chi)$. For such a formula, suppose that φ is also $(\psi' \to \chi')$. By Lemma 1.3, neither ψ nor ψ' is a proper initial segment of the other. Thus ψ is the same as ψ'. It follows that χ is the same as χ'. $\square$

Abbreviations. It will be convenient to introduce some abbreviations. Here they are.

$(\varphi \wedge \psi)$	for	$\neg(\varphi \to \neg\psi)$;
$(\varphi \vee \psi)$	for	$(\neg\varphi \to \psi)$;
$(\varphi \leftrightarrow \psi)$	for	$((\varphi \to \psi) \wedge (\psi \to \varphi))$;
$\exists x\varphi$	for	$\neg\forall x\neg\varphi$;
$x \neq y$	for	$\neg x = y$.

Bear in mind that $\vee$, $\wedge$, $\leftrightarrow$, $\exists$, and $\neq$ are not actually symbols of our languages. Given a formula abbreviated by the use of these symbols, one may eliminate the symbols via the contextual definitions just given, thus getting a genuine formula.

Parentheses convention. We will often omit parentheses where there is no ambiguity. We also adopt a convention that will allow parentheses to be omitted when there would be ambiguity without the parentheses. The convention is that omitted parentheses are grouped to the right. For example

$$(\varphi \to \psi \to \chi)$$

abbreviates

$$(\varphi \to (\psi \to \chi)).$$

Free and bound variables:

Informally, we define *occurrence* of a variable x in a formula to be *free* if it is not in the scope of a quantifier expression $\forall x$. For example, the first occurrence of v_1 in

$$\forall v_2(Pv_1v_2 \to \forall v_1 Pv_1v_1)$$

is free, while the second occurrence is *bound*.

Officially we define freedom of occurrences of variables in formulas by *recursion on length*. The definition is as follows.

(a) Every occurrence of a variable in an atomic formula is free.

(b) An occurrence of a variable x in $\neg\varphi$ is free just in case the corresponding occurrence of x in φ is free.

(c) An occurrence of a variable x in $(\varphi \to \psi)$ is free just in case the corresponding occurrence of x in φ or ψ is free.

(d) An occurrence of a variable x in $\forall y\varphi$ is free just in case in case it corresponds to a free occurrence of x in φ and x and y are different variables.

Note that clause (a) defines freedom in atomic formulas directly, and each of the other clauses defines freedom in a formula from freedom in shorter formulas.

Formula induction and formula recursion. The recursive definition of freedom can also be thought of as a definition by *formula recursion.* In each of the clause (b)-(d), freedom for a formula is defined from freedom for the formula or formulas from which it is immediately constructed. There is a corresponding notion of *proof by formula induction.* To prove by formula induction that every formula has some property P, one

(a) proves that all atomic formulas have P,

(b) proves that, for any formula φ, $\neg\varphi$ has P if φ does,

(c) proves that, for any formulas φ and ψ, $(\varphi \to \psi)$ has P if both φ and ψ do,

(d) and proves that for any formula φ and any variable x, $\forall x\varphi$ has P if φ does.

Notice he proofs given for Lemmas 1.1 and Lemma 1.3 do not work as proofs by formula induction. The two lemmas can be proved using formula induction, but the those proofs are more complex than the ones using induction on length.

Exercise 1.2. Let $\mathcal{L}$ be $\emptyset$, the language of identity. Prove by induction on length that every formula of $\mathcal{L}$ has exactly one more occurrence of the $=$ than it does of $\to$.

Exercise 1.3. Let $\mathcal{L}$ be a language in which f is a one-place function symbol, g is a two-place function symbol symbol, and h is a three-place function symbol. The term

$$hgfv_1hv_3gv_3v_3gfv_2v_1fv_1gffv_3fv_4$$

is $ht_1t_2t_3$ for some terms t_1, t_2, and t_3. What are these three terms?

2 Models, Truth, and Logical Implication

Models. A *model* $\mathfrak{A}$ for a language $\mathcal{L}$ is an ordered pair consisting of (a) a non-empty set $A = |\mathfrak{A}|$, the *universe* or *domain* of the model, and (b) a function assigning

(1) to each k-place predicate symbol P, a subset $P_{\mathfrak{A}}$ of A^k, i.e., a set of k-tuples of elements of A;

(2) to each k-place function symbol F, a function $F_{\mathfrak{A}} : A^k \to A$.

We regard A^0 as having a single member, the unique 0-tuple. This means that, for 0-place P and F, the functions $P_{\mathfrak{A}}$ and $F_{\mathfrak{A}}$ have only one value. For the predicate case, when P is a sentence symbol, we regard $P_{\mathfrak{A}}$ as a truth-value **T** or **F**. In the function case, when F is a constant, we regard $F_{\mathfrak{A}}$ is a member of A.

As a convention, when we denote a model by a Fraktur letter (which we don't always do), then we denote the universe of the model by the corresponding italic Roman letter.

We in the case of a finite language (a language with only finitely many predicate and function symbols), we sometimes specify a model as a tuple consisting of its domain and the the values of its associated function. For example, we will describe the standard model of arithmetic as $(\mathbb{N}, 0, S, +, \cdot, \leq)$, meaning that the number 0 is the unique value of F_0, the successor function is F_S, the addition function is F_+, etc.

Our next goal is do define truth in a model $\mathfrak{A}$. An immediate problem is how to handle formulas like $v_1 = c$ and Pv_1. (Here c is a constant and P is a one-place predicate symbol.) The model assigns a member $c_{\mathfrak{A}}$ to c, but it does not assign anything to the variable v_1. Thus it makes no sense to talk of the truth or falsity in the model of these two formulas.

Sentences. A *sentence* is a formula with no free occurrences of variables.

Clearly it is the *sentences* that should be true or false in $\mathfrak{A}$. Hence we might try forgetting about non-sentences and defining true for sentences by recursion on length. But this strategy does not work. It would work if, for example, every member of A were of the form $c_{\mathfrak{A}}$ for some constant c, but this does not happen in general. Consider an extreme case when the language has no constants. How are we to define truth for $\forall v_1 P v_1$? We cannot make use of the truth or falsity of for $P v_1$. That formula is neither true nor false, and furthermore it's hard to see how its truth or falsity could tell us whether $\forall v_1 P v_1$ was true or false.

Variable assignments. A *variable assignment (for a model $\mathfrak{A}$)* is a function that assigns a member of A to each variable v_i.

The solution to the problem of how to define truth is as follows. We will define a truth-value for arbitrary formulas formulas in a model *and* under a variable assignment. The truth-values of sentences will be depend only on the model and not on the variable assignment.

Let $\mathcal{L}$ be a language, let $\mathfrak{A}$ be a model, and let s be a variable assignment.

Denotation of terms. By recursion on length, we define a *denotation* $\mathrm{den}^s_{\mathfrak{A}}(t)$ for each term t of $\mathcal{L}$.

(1) For all variables x, $\mathrm{den}^s_{\mathfrak{A}}(x) = s(x)$.

(2) For all constants c, $\mathrm{den}^s_{\mathfrak{A}}(c) = c_{\mathfrak{A}}$.

(3) If t is $f t_1 \ldots t_n$ where f is an n-placed function symbol and $t_1, \ldots, t_n$ are terms, then $\mathrm{den}^s_{\mathfrak{A}}(t) = f_{\mathfrak{A}}(\mathrm{den}^s_{\mathfrak{A}}(t_1), \ldots, \mathrm{den}^s_{\mathfrak{A}}(t_n))$.

Truth-values of formulas. By recursion on length we define $\mathrm{tv}^s_{\mathfrak{A}}(\varphi)$ for each formula φ.

(1) If φ is $t_1 = t_2$ where t_1 and t_2 are terms, then $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T}$ if $\mathrm{den}^s_{\mathfrak{A}}(t_1) = \mathrm{den}^s_{\mathfrak{A}}(t_2)$, and $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{F}$ otherwise.

(2) If φ is $P t_1 \ldots t_n$ where P is an n-place predicate symbol and $t_1, \ldots, t_n$ are terms, then $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T}$ if $(\mathrm{den}^s_{\mathfrak{A}}(t_1), \ldots, \mathrm{den}^s_{\mathfrak{A}}(t_n))$ belongs to $P_{\mathfrak{A}}$, and $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{F}$ otherwise. (For $n = 0$, this means $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T} \Leftrightarrow P_{\mathfrak{A}} = \mathbf{T}$.)

(3) If φ is $\neg\psi$, then $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T}$ if $\mathrm{tv}^s_{\mathfrak{A}}(\psi) = \mathbf{F}$, and $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{F}$ otherwise.

(4) If φ is $(\psi \to \chi)$, then $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{F}$ if $\mathrm{tv}^s_{\mathfrak{A}}(\psi) = \mathbf{T}$ and $\mathrm{tv}^s_{\mathfrak{A}}(\chi) = \mathbf{F}$, and $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T}$ otherwise.

(5) If φ is $\forall x \psi$, then $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T}$ if, for all $a \in A$, if s' agrees with s except that $s'(x) = a$, then $\mathrm{tv}^{s'}_{\mathfrak{A}}(\psi) = \mathbf{T}$, and $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{F}$ otherwise.

We define a formula φ to be *true in* $\mathfrak{A}$ *under* s if $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T}$. (Often the word *satisfied* is used instead of *true*.)

Lemma 2.1. *For any model* $\mathfrak{A}$ *and any formula* φ*, if* s_1 *and* s_2 *are any two variable assignments such that* $s_1(x) = s_2(x)$ *for every variable with a free occurrence in* φ*, then* $\mathrm{tv}^{s_1}_{\mathfrak{A}}(\varphi) = \mathrm{tv}^{s_2}_{\mathfrak{A}}(\varphi)$*.*

Exercise 2.1. Prove Lemma 2.1. To do this, fix $\mathfrak{A}$ and prove by induction on length that every formula φ has the property that the Lemma say it has.

Truth in a model. By Lemma 2.1, we may define $\mathrm{tv}_{\mathfrak{A}}(\sigma)$ for sentences σ by setting

$$\mathrm{tv}_{\mathfrak{A}}(\sigma) = \mathrm{tv}^s_{\mathfrak{A}}(\sigma),$$

where s is any variable assignment. We define a sentence σ to be *true in* $\mathfrak{A}$ if $\mathrm{tv}^s_{\mathfrak{A}}(\sigma) = \mathbf{T}$.

Truth of sets of formulas and sets of sentences. We say that a set Γ of formulas is *true in* $\mathfrak{A}$ *under* s if all the formulas in Γ are true in $\mathfrak{A}$ under s. Similarly, we say that a set Σ of sentences is *true in* $\mathfrak{A}$ if all the sentences in Σ are true in $\mathfrak{A}$.

It is not hard to see, using the definition of truth and the contextual definition of $\exists x$, that $\mathrm{tv}^s_{\mathfrak{A}}(\exists x\psi) = \mathbf{T}$ if and only if there is an $a \in A$ such that, if s' agrees with s except that $s'(x) = a$, then $\mathrm{tv}^s_{\mathfrak{A}}(\psi) = \mathbf{T}$.

Exercise 2.2. Let $\mathcal{L} = \{c, p, P, Q, f\}$, where c is a constant, p is a 0-place predicate symbol, P is a one-place predicate symbol, Q is a two-place predicate symbol, and f is a one-place function symbol. Let $\mathfrak{A}$ be the following model for $\mathcal{L}$.

$$\begin{array}{rcl} A & = & \{d_1, d_2\} \\ c_{\mathfrak{A}} & = & d_2 \\ p & = & \mathbf{T} \\ P_{\mathfrak{A}} & = & \{d_1\} \\ Q_{\mathfrak{A}} & = & \{(d_1, d_2), (d_2, d_2)\} \\ f_{\mathfrak{A}}(d_1) & = & d_1 \\ f_{\mathfrak{A}}(d_2) & = & d_1 \end{array}$$

Here d_1 and d_2 are distinct objects.

Which of the following sentences are true in $\mathfrak{A}$? Explain your answers.

$$\begin{array}{ll} (a)\ \exists v_1 \forall v_2\, Q v_2 v_1 & (b)\ \forall v_1 (P v_1 \vee Q c v_1) \\ (c)\ \forall v_1 (P v_1 \to p) & (d)\ \exists v_1 (P v_1 \to p) \\ (e)\ \exists v_1 \forall v_2 f v_2 = v_1 & (f)\ \forall v_1 \exists v_2 Q f v_1 v_2 \end{array}$$

Logical implication:

If Γ is a set of formulas and φ is a formula, then we say that Γ *logically implies* φ (in symbols, $\Gamma \models \varphi$) if and only if, for every model $\mathfrak{A}$ and every variable assignment s,

$$\text{if } \Gamma \text{ is true in } \mathfrak{A} \text{ under } s, \text{ then } \varphi \text{ is true in } \mathfrak{A} \text{ under } s\,.$$

A formula or set of formulas is *valid* if it is true in every model under every variable assignment; it is *satisfiable* if it is true in some model under some variable assignment. A formula φ is valid if and only if $\emptyset \models \varphi$, and we abbreviate $\emptyset \models \varphi$ by $\models \varphi$. We shall be interested in the notions of logical implication, validity, and satisfiability mainly for sets of sentences and sentences. In this case variable assignments s play no role. A set Σ of sentences logically implies a sentence φ if and only if, for every model $\mathfrak{A}$,

$$\text{if } \Sigma \text{ is true in } \mathfrak{A}, \text{ then } \varphi \text{ is true in } \mathfrak{A}\,.$$

Exercise 2.3. Let $\mathcal{L} = \{P, Q, c_1, c_2, f\}$, where P is a one-place predicate symbol, let Q is a two-place predicate symbol, c_1 and c_2 are constants, and f is a two-place function symbol. For each of the following pairs (Γ, φ), tell whether $\Gamma \models \varphi$. If the answer is yes, explain why. If the answer is no, then describe a model or a model and a variable assignment showing that the answer is no.

(a) Γ: $\{\forall v_1 \exists v_2\, Q v_1 v_2\}$; φ: $\exists v_2 \forall v_1\, Q v_1 v_2$.

(b) Γ: $\{\exists v_1 \forall v_2\, Q v_1 v_2\}$; φ: $\forall v_2 \exists v_1\, Q v_1 v_2$.

(c) Γ: $\{\forall v_1\ \ Q v_1 v_1, Q c_1 c_2\}$; φ: $Q c_2 c_1$;

(d) Γ: $\{\forall v_1 \forall v_2\, Q v_1 v_2\}$; φ: $\forall v_2 \forall v_1\, Q v_1 v_2$;

(e) Γ: $\{P v_1\}$; φ: $\forall v_1\, P v_1$.

(f) Γ: $\{\forall v_1 f v_1 c_1 \neq f v_1 v_1\}$; φ: $f c_1 c_2 \neq c_2$.

Exercise 2.4. Let $\mathcal{L} = \{P\}$, where P is a two-place predicate symbol. Describe a model in which all three of the following sentences are all true.

(a) $\forall v_1 \exists v_2\, P v_1 v_2$.

(b) $\forall v_1 \forall v_2 (P v_1 v_2 \to \neg P v_2 v_1)$.

(c) $\forall v_1 \forall v_2 \forall v_3 ((P v_1 v_2 \wedge P v_2 v_3) \to P v_1 v_3)$.

Can these three sentences be true in a model whose universe is finite? Explain.

Exercise 2.5. Let $\mathcal{L} = \{f\}$, where f is a one-place function symbol. Does $\{\forall v_1 \forall v_2 (f(v_1) = f(v_2) \to v_1 = v_2)\}$ logically imply $\forall v_1 \exists v_2 f(v_2) = v_1$? If the answer is yes, explain why. If it is no, describe a model showing this.

Exercise 2.6. Let Γ and Δ be sets of formulas of some language $\mathcal{L}$ let φ, ψ, and $\varphi_1, \ldots, \varphi_n$ be formulas of $\mathcal{L}$, and let p be a sentence symbol of $\mathcal{L}$. Prove each of the following.

(1) $\Gamma \cup \{\varphi\} \models \psi$ if and only if $\Gamma \models (\varphi \to \psi)$.

(2) $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\} \models \psi$ if and only if $\Gamma \models (\varphi_1 \to \cdots \to \varphi_n \to \psi)$.

(3) Γ is satisfiable if and only if $\Gamma \not\models (p \wedge \neg p)$.

(4) If $\Gamma \models$ every formula belonging to Δ and if $\Delta \models \psi$, then $\Gamma \models \psi$.

Here have used the convention announced on page 6 that omitted parentheses group to the right.

Exercise 2.7. There is an important relation between satisfiability and logical implication.

$$\Gamma \models \varphi \text{ if and only if } \Gamma \cup \{\neg\varphi\} \text{ is not satisfiable.}$$

Prove that this relation obtains.

Sentential implication and tautologies.

A formula of is *prime* if it is either atomic or of the form $\forall x \varphi$. Equivalently, a formula is prime if it is not of the form $\neg\varphi$ or of the form $(\varphi \to \psi)$.

Examples:

- $fv_1 = c$ and $\forall v_1(Pv_1 \to Qcv2)$ are prime.
- $(Pv_2 \to \neg Pv_3)$ and $\exists v_1 Pv_1$ are not prime. ($\exists v_1 Pv_1$ is really the formula $\neg\forall v_1 \neg Pv_1$.)

Prime formula valuations. Fix a language $\mathcal{L}$. A *prime formula valuation for $\mathcal{L}$* (is a function v that assigns a truth-value $\mathbf{T}$ or $\mathbf{F}$ to each prime formula of $\mathcal{L}$. Prime formula valuations are sometimes called *truth-value assignments to the prime formulas* or *extended valuations.*

Let v be a prime formula valuation for a language $\mathcal{L}$. By recursion on length, we extend v to a function v^* that assigns a truth-value to every formula of $\mathcal{L}$.

(i) If φ is prime, then $v^*(\varphi) = v(\varphi)$.

(ii) If φ is $\neg\psi$, then $v^*(\varphi) = \mathbf{T}$ just in case $v^*(\psi) = \mathbf{F}$.

(iii) If φ is $(\psi \to \chi)$, then $v^*(\varphi) = \mathbf{F}$ just in case $v^*(\psi) = \mathbf{T}$ and $v^*(\chi) = \mathbf{F}$.

A formula φ is *true under v* if $v^*(\varphi) = \mathbf{T}$. A set of formulas is *true under v* if all the formulas in the set are true under v.

Tautological implication. A set Γ of formulas *tautologically implies* a formula φ (in symbols, $\Gamma \models_{\mathrm{t}} \varphi$) if φ is true under every prime formula valuation under which Γ is true.

Tautologies. A formula φ is a *tautology* if φ is true under every prime formula valuation. Note that φ is a tautology if and only if $\emptyset \models_{\mathrm{t}} \varphi$ (which we abbreviate $\models_{\mathrm{t}} (\varphi)$.

Examples:

- $(Pv_1 \to Pv_1)$ and $(Pv_1 \to \neg\neg Pv_1)$ are tautologies.
- $\forall v_1(Pv_1 \to Pv_1)$ and $(\forall v_1 Pv_1 \to \exists v_1 Pv_1)$ are not tautologies.

Lemma 2.2. *If $\Gamma \models_{\mathrm{t}} \varphi$ then $\Gamma \models \varphi$. Hence every tautology is valid.*

Proof. Assume that $\Gamma \models_{\mathrm{t}} \varphi$. Let $\mathfrak{A}$ be a model and let s be a variable assignment. Define a prime formula valuation v by setting

$$v(\varphi) = \mathrm{tv}^s_{\mathfrak{A}}(\varphi)$$

for each prime formula φ. It follows by induction on length that

$$v^*(\varphi) = \mathrm{tv}^s_{\mathfrak{A}}(\varphi)$$

for every formula φ. Assume that Γ is true in $\mathfrak{A}$ under s. Then Γ is true under v. Since $\Gamma \models_{\mathrm{t}} \varphi$, $v^*(\varphi) = \mathbf{T}$. Hence $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T}$. $\square$

3 Formal Deduction

For each language $\mathcal{L}$, we define a *deductive system* for $\mathcal{L}$.

In describing the system, we need the following definition. If φ is a formula, x is a variable, and t is a term, then $\varphi(x;t)$ is the formula that results by replacing each free occurrence of x in φ by an occurrence of t.

Axioms:

(1) All tautologies.

(2) Identity Axioms:

  (a) $t = t$
  for all terms t;

  (b) $t_1 = t_2 \rightarrow (\varphi(x;t_1) \rightarrow \varphi(x;t_2))$
  for all terms t_1 and t_2, all variables x, and all formulas φ such that there is no variable y occurring in t_1 or t_2 such that there is free occurrence of x in φ in a subformula of φ of the form $\forall y\psi$.

(3) Quantifier Axioms:

$$\forall x\varphi \rightarrow \varphi(x;t)$$

for all formulas φ, variables x, and terms t such that there is no variable y occurring in t such that there is a free occurrence of x in φ in a subformula of φ of the form $\forall y\psi$.

Rules of Inference:

Modus Ponens (MP) $\quad \dfrac{\varphi\,,\ (\varphi \rightarrow \psi)}{\psi}$

Quantifier Rule (QR) $\quad \dfrac{(\varphi \rightarrow \psi)}{(\varphi \rightarrow \forall x\psi)}$

provided the variable x does not occur free in φ.

Discussion of some of the axioms and rules.

(1) Identity Axiom Schema (a) is self-explanatory. Schema (b) is a formal version of the *Indiscernibility of Identicals*, also called *Leibniz's Law.*

(2) The Quantifer Axiom Schema is often called the schema of *Universal Instantiation*. Its idea is that whatever is true of a all objects in the domain is true of whatever object t might denote. The reason for the odd-looking restriction is that instances where the restriction fails do not conform to the idea. Here is an example. Let φ be $\exists v_2\, v_1 \neq v_2$, let x be v_1 and let t be v_2. The instance of the schema would be

$$\forall v_1 \exists v_2\, v_1 \neq v_2 \rightarrow \exists v_2\, v_2 \neq v_2.$$

The antecedent is true in all models whose domains have more than one element, but the consequent is not satisfiable.

(3) As we will explain later, the Quantifier Rule is not a valid rule. The reason it will be legitimate for us to use it as a rule is that we will allow only sentences as premises of our deductions. How this works will be explained in the proof of the Soundness Theorem.

Deductions. A *deduction* in $\mathcal{L}$ from a set Σ of sentences is a finite sequence $\mathbf{D}$ of formulas such that whenever a formula φ occurs in the sequence $\mathbf{D}$ then at least one of the following holds.

(1) $\varphi \in \Sigma$.

(2) φ is an axiom.

(3) φ follows by Modus Ponens from two formulas occurring earlier in the sequence $\mathbf{D}$ or follows by the Quantifier Rule from a formula occurring earlier in $\mathbf{D}$.

A *deduction in $\mathcal{L}$ of a formula φ from a set* Σ of sentences is a deduction $\mathbf{D}$ in $\mathcal{L}$ from Σ with φ on the last line of $\mathbf{D}$. We write $\Sigma \vdash_{\mathcal{L}} \varphi$ and say φ is *deducible* in $\mathcal{L}$ from Σ to mean that there is a deduction in $\mathcal{L}$ of φ from Σ. We write $\vdash_{\mathcal{L}} \varphi$ for $\emptyset \vdash_{\mathcal{L}} \varphi$.

Remark. Unless two or more languages are in play, we will omit the subscript $\mathcal{L}$.

In order to avoid dealing directly with long formulas and long deductions, it will be useful to begin by justifying some derived rules.

Lemma 3.1. *Assume that $\Sigma \vdash \varphi_i$ for $1 \leq i \leq n$ and $\{\varphi_1, \ldots, \varphi_n\} \models_{\mathrm{t}} \psi$. Then $\Sigma \vdash \psi$. (See page 13 for the definition of $\models_{\mathrm{t}}$.)*

Proof. If we string together deductions witnessing that $\Sigma \vdash \varphi_i$ for each i, then we get a deduction from Σ in which each φ_i is a line. The fact that $\{\varphi_1, \dots, \varphi_n\} \models_t \psi$ gives us that the formula

$$(\varphi_1 \rightarrow \varphi_2 \rightarrow \cdots \rightarrow \varphi_n \rightarrow \psi)$$

is a tautology. Appending this formula to our deduction and applying MP n times, we get ψ. □

Lemma 3.1 justifies a derived rule, which we call SL. A formula ψ follows from formulas $\varphi_1, \dots, \varphi_n$ by SL iff

$$\{\varphi_1, \dots, \varphi_n\} \models_t \psi.$$

Lemma 3.2. *If $\Sigma \vdash \varphi$ then $\Sigma \vdash \forall x\varphi$ (for any variable x).*

Proof. Assume that $\Sigma \vdash \varphi$. Begin with a deduction from Σ with last line φ. Let $\top$ be the formula

$$(\exists v_1\, v_1 = v_1 \vee \neg\exists v_1\, v_1 = v_1).$$

Use SL to get the line $(\top \rightarrow \varphi)$. Now apply QR to get $(\top \rightarrow \forall x\varphi)$. Finally use SL to get $\forall x\varphi$. □

Remark. Any tautology that is a formula of the language $\mathcal{L}$ under consideration and does not contain a free occurrence of x would do in place of $\top$. The formula $\top$ has the additional property of being a *sentence* of *every* language $\mathcal{L}$.

Lemma 3.2 justifies a derived rule, which we call Gen:

$$\text{Gen} \qquad \frac{\varphi}{\forall x\varphi}$$

Lemma 3.3. *For all formulas φ and ψ,*

$$\vdash \forall x(\varphi \rightarrow \psi) \rightarrow (\forall x\varphi \rightarrow \forall x\psi).$$

Proof. Here is an abbreviated deduction.

1.	$\forall x(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \psi)$	QAx
2.	$\forall x\varphi \rightarrow \varphi$	QAx
3.	$(\forall x(\varphi \rightarrow \psi) \wedge \forall x\varphi) \rightarrow \psi$	1,2; SL
4.	$(\forall x(\varphi \rightarrow \psi) \wedge \forall x\varphi) \rightarrow \forall x\psi$	3; QR
5.	$\forall x(\varphi \rightarrow \psi) \rightarrow (\forall x\varphi \rightarrow \forall x\psi)$	4; SL

□

Exercise 3.1. Show that $\vdash (\exists v_1\, Pv_1 \to \exists v_2\, Pv_2)$.

Exercise 3.2. Show that $\{\forall v_1\, Pv_1\} \vdash \exists v_1\, Pv_1$.

Lemma 3.4. *If* $\Sigma \vdash (\varphi \to \psi)$ *then* $\Sigma \vdash (\forall x\varphi \to \forall x\psi)$.

Proof. Start with a deduction from Σ with last line $(\varphi \to \psi)$. Use Gen to get the line $\forall x(\varphi \to \psi)$. Then apply Lemma 3.3 and MP. $\square$

Theorem 3.5 (Deduction Theorem). *Let* Σ *be a set of sentences, let* σ *be a sentence, and let* φ *be a formula. If* $\Sigma \cup \{\sigma\} \vdash \varphi$ *then* $\Sigma \vdash (\sigma \to \varphi)$.

Proof. Assume that $\Sigma \cup \{\sigma\} \vdash \varphi$. Let **D** be a deduction of φ from $\Sigma \cup \{\sigma\}$. We prove that

$$\Sigma \vdash (\sigma \to \psi)$$

for every line ψ of **D**. Assume that this is false. Consider the first line ψ of **D** such that $\Sigma \not\vdash (\varphi \to \psi)$.

Assume first that ψ either belongs to Σ or is an axiom. Then $\Sigma \vdash \psi$ and $(\varphi \to \psi)$ follows from ψ by SL. Hence $\Sigma \vdash (\varphi \to \psi)$.

Assume next that ψ is φ. Since $(\varphi \to \varphi)$ is a tautology, $\Sigma \vdash (\varphi \to \varphi)$.

Assume next that ψ follows by MP from formulas χ and $(\chi \to \psi)$ on earlier lines of **D**. Since ψ is the first "bad" line of **D**, $\Sigma \vdash (\sigma \to \chi)$ and $\Sigma \vdash (\sigma \to (\chi \to \psi))$. Since

$$\{(\sigma \to \chi), (\sigma \to (\chi \to \psi))\} \models_{\text{sl}} (\sigma \to \psi),$$

Lemma 3.1 gives us that $\Sigma \vdash (\sigma \to \psi)$.

Finally assume that ψ is $(\chi \to \forall x\rho)$ and that ψ follows by QR from an earlier line $(\chi \to \rho)$ of **D**. Since ψ is the first "bad" line of **D**, $\Sigma \vdash (\sigma \to (\chi \to \rho))$. Starting with a deduction from Σ of $(\sigma \to (\chi \to \rho))$, we can get a deduction from Σ of $(\sigma \to (\chi \to \forall x\rho))$ as follows.

$\cdots$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\cdots$
$\cdots$	$\cdots$	$\cdots$
n	$\sigma \to (\chi \to \rho)$	$\cdots$
$n+1.$	$(\sigma \wedge \chi) \to \rho$	n; SL
$n+2.$	$(\sigma \wedge \chi) \to \forall x\rho$	$n+1$; QR
$n+3.$	$\sigma \to (\chi \to \forall x\rho)$	$n+2$; SL

Note that the variable x has no free occurrences in σ because σ is a sentence, and we know that it has no free occurrences in χ because we know that QR was used in $\mathbf{D}$ to get $\chi \to \forall x\rho$ from $\chi \to \rho$.

This contradiction completes the proof that the "bad" line C cannot exist. Applying this fact to the last line of $\mathbf{D}$, we get that $\Sigma \vdash (\sigma \to \varphi)$. $\square$

The Deduction Theorem is useful in showing that conditionals are deducible. If σ is a sentence, then to show $\Sigma \vdash (\sigma \to \varphi)$ it is enough to show that $\Sigma \cup \{\sigma\} \vdash \varphi$.

Exercise 3.3. Show that, for any variable x and constant c,

$$\vdash (Pc \leftrightarrow \forall x(x = c \to Px)).$$

(See page 5 for the contextual definition of $\leftrightarrow$.)

Hint. Show that $\vdash (Pc \to \forall x(x = y \to Px))$ and $\vdash (\forall x(x = c \to Px) \to Pc)$ and then use SL. In showing that the two conditionals are deducible, use the Deduction Theorem.

Consistency. A set Σ of sentences of $\mathcal{L}$ is *inconsistent* in $\mathcal{L}$ if there is a formula ψ such that $\Sigma \vdash_{\mathcal{L}} \psi$ and $\Sigma \vdash_{\mathcal{L}} \neg\psi$. Otherwise Σ is *consistent.*

Theorem 3.6. *Let Σ and Δ be sets of sentences, let σ and $\sigma_1, \ldots, \sigma_n$ be sentences, and let φ be a formula.*

(1) $\Sigma \cup \{\sigma\} \vdash \varphi$ if and only if $\Sigma \vdash (\sigma \to \varphi)$.

(2) $\Sigma \cup \{\sigma_1, \ldots, \sigma_n\} \vdash \varphi$ if and only if $\Sigma \vdash (\sigma_1 \to \ldots \to \sigma_n \to \varphi)$.

(3) Σ is consistent if and only if there is some formula χ such that $\Sigma \nvdash \chi$.

(4) If $\Sigma \vdash$ every formula $\in \Delta$ and if $\Delta \vdash \varphi$, then $\Sigma \vdash \varphi$.

Proof. (1) The "only if" direction is the Deduction Theorem. The "if" direction is the Deduction Theorem's converse. To prove the "if" direction, note that any deduction of $(\sigma \to \varphi)$ from Σ can be turned into a deduction of φ from $\Sigma \cup \{\sigma\}$ by adding the lines σ and φ, the latter line coming by MP.

(2) For the "only if" direction, apply the Deduction Theorem n times; for the "if" direction, apply the converse of the Deduction Theorem n times.

(3) The "only if" direction is obvious. For the "if" direction, we prove the contrapositive. Assume that Σ is inconsistent. Let ψ be a formula such that $\Sigma \vdash \psi$ and $\sigma \vdash \neg\psi$. For any formula χ,

$$\{\psi, \neg\psi\} \models_{\mathrm{t}} \chi.$$

Hence $\Sigma \vdash \chi$ by SL.

(4) Let $\mathbf{D}$ be a deduction of φ from Δ. Let $\tau_1, \ldots, \tau_n$ be all the members of Δ that appear as lines of $\mathbf{D}$. For each i, let $\mathbf{D}_i$ be a deduction of τ_i from Σ. To get a deduction of φ from Σ, put the D_i end to end and follow them by $\mathbf{D}$. $\square$

Lemma 3.7. *For all formulas φ and any variables x and y,*

$$\vdash \exists x \forall y \varphi \to \forall y \exists x \varphi.$$

Proof. Here is an abbreviated deduction.

1.	$\forall y\varphi \to \varphi$	QAx
2.	$\neg\varphi \to \neg\forall y\varphi$	1; SL
3.	$\forall x\neg\varphi \to \neg\varphi$	QAx
4.	$\forall x\neg\varphi \to \neg\forall y\varphi$	2,3; SL
5.	$\forall x\neg\varphi \to \forall x\neg\forall y\varphi$	4; QR
6.	$\neg\forall x\neg\forall y\varphi \to \neg\forall x\neg\varphi$ [$\exists x\forall y\varphi \to \exists x\varphi$]	5; SL
7.	$\exists x\forall y\varphi \to \forall y\exists x\varphi$	6; QR

$\square$

Exercise 3.4. Show that

$$\{\forall v_1 \forall v_2 (P v_1 v_2 \lor P v_2 v_1)\} \ \vdash \ \forall v_1 P v_1 v_1.$$

Exercise 3.5. Show that

$$\vdash \ \forall v_1 \exists v_2 f v_1 = v_2.$$

Here f is a one-place function symbol.

Exercise 3.6. Let c_1 and c_2 be constants. Show that

$$\{c_1 = c_2\} \ \vdash \ c_2 = c_1.$$

4 Soundness and Completeness

Soundness and completeness. A system **S** of deduction for $\mathcal{L}$ is *sound* if, for all sets Σ of sentences and all formulas φ, if $\Sigma \vdash_{\mathbf{S}} \varphi$ then $\Sigma \models \varphi$. A system **S** of deduction for $\mathcal{L}$ is *complete* if, for all sets Σ of sentences and all formulas φ, if $\Sigma \models \varphi$ then $\Sigma \vdash_{\mathbf{S}} \varphi$. In this section, we will prove that our systems of deduction for languages $\mathcal{L}$ are all sound and complete.

Exercise 4.1. Prove that all instances of Identity Axiom Schema (b) are valid.

Exercise 4.2. Prove that all instances of the Quantifier Axiom Schema are valid.

Hint for Exercises 4.1 and 4.2. For terms t^* and t and variables x, let $t^*(x;t)$ be the result of replacing the occurrences of x in t^* by occurrences of t.

Let $\mathfrak{A}$ be a model and and let s be variable assignment. Let x be variable and let t be a term. Assume that $s(x) = \mathrm{den}^s_{\mathfrak{A}}(t)$. Prove by induction on length that, for all terms t^*,

$$\mathrm{den}^s_{\mathfrak{A}}(t^*) = \mathrm{den}^s_{\mathfrak{A}}(t^*(x;t)).$$

Next prove by induction on length that, for all formulas φ, if φ, x, and t satisfy the restriction in the statement of the Quantifier Axiom Schema, then

$$v^s_{\mathfrak{A}}(\varphi) = v^s_{\mathfrak{A}}(\varphi;t).$$

To show that an instance $t_1 = t_2 \to \varphi(x;t_1) \to \varphi(x;t_2))$ of Identity Axiom Schema (b) is is true in $\mathfrak{A}$ under s, apply what you have proved to the terms t_1 and t_2.

Theorem 4.1 (Soundness). *For each $\mathcal{L}$, our system of deduction for $\mathcal{L}$ is sound.*

Proof. Let **D** be a deduction in $\mathcal{L}$ of a formula φ from a set Σ of sentences. We will show that, for every line ψ of **D**, $\Sigma \models \psi$. Applying this to the last line of **D**, this will give us that $\Sigma \models \varphi$.

Assume that what we wish to show is false. Let ψ be the first line of **D** such that $\Sigma \not\models \psi$.

Assume first that ψ is an axiom. Using Exercises 4.1 and 4.2, it is easy to see that all the axioms are valid. Hence $\models \psi$ and so $\Sigma \models \psi$.

Assume next that $\psi \in \Sigma$. Trivially $\Sigma \models \psi$.

Assume next that ψ follows by MP from formulas χ and $(\chi \rightarrow \psi)$ on earlier lines of **D**. Since ψ is the first "bad" line of **D**, $\Sigma \models \chi$ and $\Sigma \models (\chi \rightarrow \psi)$. It follows that $\Sigma \models \psi$.

Finally assume that ψ is $(\chi \rightarrow \forall x\rho)$ and that ψ follows by QR from an earlier line $(\chi \rightarrow \rho)$ of **D**. Since ψ is the first "bad" line of **D**, $\Sigma \models (\chi \rightarrow \rho)$. Let $\mathfrak{A}$ be any model and let s be any variable assignment. We assume that $v^s_{\mathfrak{A}}(\Sigma) = \mathbf{T}$ (by which we mean that $v^s_{\mathfrak{A}}(\pi) = \mathbf{T}$ for each $\pi \in \Sigma$), and we show that $v^s_{\mathfrak{A}}(\chi \rightarrow \forall x\rho) = \mathbf{T}$. To do this, we assume that $v^s_{\mathfrak{A}}(\chi) = \mathbf{T}$ and we show that $v^s_{\mathfrak{A}}(\forall x\rho) = \mathbf{T}$. Let a be any element of A and let s' be any variable assignment that agrees with s except that $s'(x) = a$. We must show that $v^{s'}_{\mathfrak{A}}(\rho) = \mathbf{T}$. Since Σ is a set of sentences, $v^{s'}_{\mathfrak{A}}(\Sigma) = v^s_{\mathfrak{A}}(\Sigma) = \mathbf{T}$. Since the variable x does not occur free in χ, $v^{s'}_{\mathfrak{A}}(\chi) = v^s_{\mathfrak{A}}(\chi) = \mathbf{T}$. Since $\Sigma \models (\chi \rightarrow \rho)$, it follows that $v^{s'}_{\mathfrak{A}}(\rho) = \mathbf{T}$ □

We now begin the proof of the completeness of our deductive systems. The following fact will be be used in the proof.

Exercise 4.3. Our system of deduction for a language $\mathcal{L}$ is complete if and only if every set of sentences consistent in $\mathcal{L}$ is satisfiable in $\mathcal{L}$.

Lemma 4.2. *Let Σ be a set of sentences of a language $\mathcal{L}$ consistent in $\mathcal{L}$, and let σ be a sentence of $\mathcal{L}$. Either $\Sigma \cup \{\sigma\}$ is consistent in $\mathcal{L}$ or $\Sigma \cup \{\neg\sigma\}$ is consistent in $\mathcal{L}$.*

Proof. Assume for a contradiction neither $\Sigma \cup \{\sigma\}$ nor $\Sigma \cup \{\neg\sigma\}$ is consistent. It follows that there are formulas ψ and ψ' such that

(i) $\Sigma \cup \{\sigma\} \vdash \psi$;

(ii) $\Sigma \cup \{\sigma\} \vdash \neg\psi$;

(iii) $\Sigma \cup \{\neg\sigma\} \vdash \psi'$;

(iv) $\Sigma \cup \{\neg\sigma\} \vdash \neg\psi'$.

Using SL and the with (iii), (iv), and the Deduction Theorem, we can show that $\Sigma \vdash \sigma$. This fact, together with (i), (ii), and the Deduction Theorem, allows us to show that $\Sigma \vdash \psi$ and $\Sigma \vdash \neg\psi$. Thus we have the contradiction that Σ is inconsistent. □

Simplifying assumption. From now on consider only languages $\mathcal{L}$ that are *countable*, i.e., whose predicate and function symbols can be arranged in a finite or infinite list. Most of the facts we will prove can be proved without this without this restriction, but the proofs involve concepts beyond the scope of this course.

Henkin sets. A set Σ of sentences in a language $\mathcal{L}$ is *Henkin in* $\mathcal{L}$ if, for each formula φ of $\mathcal{L}$ and each variable x, if (i) below holds, then (ii) also holds.

(i) $\varphi(x; c) \in \Sigma$ for all constants of $\mathcal{L}$.

(ii) $\forall x \varphi \in \Sigma$.

Lemma 4.3. *Let Σ be set of sentences of a language $\mathcal{L}$ consistent in $\mathcal{L}$. Let $\mathcal{L}^*$ be gotten from $\mathcal{L}$ by adding infinitely many new constants. There is a set Σ^* of sentences of $\mathcal{L}^*$ such that*

(1) $\Sigma \subseteq \Sigma^*$;

(2) Σ^* is consistent in $\mathcal{L}^*$;

(3) for every sentence σ of $\mathcal{L}^*$, either σ belongs to Σ^* or $\neg\sigma$ belongs to Σ^*;

(4) Σ^* is Henkin.

In the proof of the lemma, we will use the following alternative characterization of the Henkin property.

Exercise 4.4. Call a set Σ of sentences in a language $\mathcal{L}$ *Henkin′ in* $\mathcal{L}$ if, for each formula φ and each variable x, if (iii) below holds, then (iv) also holds.

(iii) $\exists x \varphi \in \Sigma$.

(iv) $\varphi(x; c) \in \Sigma$ for some constant c of $\mathcal{L}$.

Let Σ^* be a set of sentences in a language $\mathcal{L}^*$ having properties (2) and (3) described in the statement of Lemma 4.3. Show that Σ^* is Henkin in $\mathcal{L}^*$ if and only if it is Henkin′ in $\mathcal{L}^*$.

Proof of Lemma 4.3. By our simplifying assumption, we have a finite or infinite list of all the predicate and function symbols of $\mathcal{L}^*$. (Recall that constants are 0-place function symbols.) Think of all the symbols of $\mathcal{L}$ as forming an infinite "alphabet" with the alphabetical order given as follows.

(i) The alphabet begins with $\neg$, $\to$, (,), $\forall$, $=$.

(ii) Next come the variables, $v_1, v_2, v_3, \ldots$.

(iii) Last come the predicate and function symbols, in the order of our given list.

Now we form an infinite list of all the sentences of $\mathcal{L}^*$. First list in alphabetical order all the (finitely many) formulas that have length 1 and that contain no variables other than v_1 and no predicate or function symbols other than the first one (in the list). Next list in alphabetical order all the remaining sentences that have length ≤ 2 and that contain no variables other than v_1 and v_2 and no predicate or function symbols other than the first two. Next list in alphabetical order all the remaining sentences that have length ≤ 3 and that contain no variables other than v_1, v_2, and v_3 and no predicate or function symbols other than the first three. Continue in this way. (If we gave the details, what we would be doing in describing this list would be to define a function by recursion on natural numbers—the function that assigns to n the formula called σ_n in following paragraph.)

Let the formulas of $\mathcal{L}^*$, in the order listed, be

$$\sigma_0,\ \sigma_1,\ \sigma_2,\ \sigma_3,\ \ldots.$$

Let

$$c_0, c_1, c_2, \ldots$$

be all the constants of $\mathcal{L}^*$.

We define, by recursion on natural numbers, a function that associates with each natural number n a set Σ_n of sentences.

We begin by setting $\Sigma_0 = \Sigma$.

Since Σ_0 is a set of sentences of $\mathcal{L}$, it contains none of the new constants added in going from $\mathcal{L}$ to $\mathcal{L}^*$. We will make sure that for each n at most two sentences belong to Σ_{n+1} but not to Σ_n. If follows that for each n only finitely many of the new constants occur in sentences in Γ_n.

We will also make sure that $\Sigma_n \subseteq \Sigma_{n+1}$ for each n.

We define Σ_{n+1} from Σ_n in two steps. For the first step, let

$$\Sigma'_n = \begin{cases} \Sigma_n \cup \{\sigma_n\} & \text{if } \Sigma_n \cup \{\sigma_n\} \text{ is consistent in } \mathcal{L}^*; \\ \Sigma_n \cup \{\neg\sigma_n\} & \text{otherwise.} \end{cases}$$

Let $\Sigma_{n+1} = \Sigma'_n$ unless both of the following hold.

(a) $\sigma_n \in \Sigma'_n$.

(b) σ_n is $\exists x_n \varphi_n$ for some variable x_n and formula φ_n.

Suppose that both (a) and (b) hold. Let i_n be the least i such that the constant c_i does not occur in any formula belonging to Σ'_n. Such an i must exist, since only finitely many of the infinitely many new constants occur in sentences in Σ'_n. Let

$$\Sigma_{n+1} = \Sigma'_n \cup \{\varphi_n(x_n; c_{i_n})\}.$$

Let $\Sigma^* = \bigcup_n \Sigma_n$.

Because $\Sigma = \Sigma_0 \subseteq \Sigma^*$, Σ^* has property (1).

We prove by mathematical induction that Σ_n is consistent in L* for each n.

Σ_0 (i.e., Σ) is consistent in $\mathcal{L}$ by hypothesis, but we must prove that it is consistent in $\mathcal{L}^*$. If Σ is inconsistent in $\mathcal{L}^*$, then by part (3) of Theorem 3.6, every formula of $\mathcal{L}^*$ is deducible from Σ in $\mathcal{L}^*$. In particular, there is a sentence τ of $\mathcal{L}$ such that both τ and its negation are both deducible from Σ in $\mathcal{L}^*$. Observe that any deduction $\mathbf{D}$ from Σ in $\mathcal{L}^*$ of a formula of $\mathcal{L}$ can be turned into a deduction from Σ in $\mathcal{L}$ of the same formula: just replace the new constants occurring in $\mathbf{D}$ by distinct variables that do not occur in $\mathbf{D}$. It follows easily that Σ is inconsistent in $\mathcal{L}$ if it is inconsistent in $\mathcal{L}^*$.

For the rest of the proof of the lemma, "consistent" means "consistent in $\mathcal{L}^*$." Assume that Σ_n is consistent. We must show that Σ_{n+1} is consistent. Lemma 4.2 implies that Σ'_n is consistent. If $\Sigma_{n+1} = \Sigma'_n$, then Σ_{n+1} is consistent. Assume then that $\Sigma_{n+1} = \Sigma'_n \cup \{\varphi_n(x_n; c_{i_n})\}$ and, in order to derive a contradiction, assume that Σ_{n+1} is inconsistent. Arguing as we did in the preceding paragraph, we get that there is a sentence of τ of $\mathcal{L}$ such that τ and $\neg\tau$ are both deducible from Σ_{n+1}:

$$\Sigma_{n+1} \vdash_{\mathcal{L}^*} (\tau \wedge \neg\tau).$$

In other words,

$$\Sigma'_n \cup \{\varphi_n(x_n; c_{i_n})\} \ \vdash_{\mathcal{L}^*} \ (\tau \wedge \neg\tau).$$

By the Deduction Theorem,

$$\Sigma'_n \vdash_{\mathcal{L}^*} \ (\varphi_n(x_n; c_{i_n}) \to (\tau \wedge \neg\tau)).$$

Let $\mathbf{D}$ be a deduction from Σ'_n in $\mathcal{L}^*$ with last line $(\varphi_n(x_n; c_{i_n}) \to (\tau \wedge \neg\tau))$. Let y be a variable not occurring in $\mathbf{D}$. Let $\mathbf{D'}$ come from d by replacing every occurrence of c_{i_n} by an occurrence of y. Since c_{i_n} does not occur Σ'_n or in φ_n, $\mathbf{D'}$ is a deduction from Σ'_n in $\mathcal{L}^*$ with last line $(\varphi_n(x_n; y) \to (\tau \wedge \neg\tau))$. We can turn $\mathbf{D'}$ into a deduction from Σ'_n in $\mathcal{L}^*$ with last line $(\exists x_n \varphi_n \to (\tau \wedge \neg\tau))$ as follows.

$\dots$	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$
$m.$	$\varphi_n(x_n; y) \to (\tau \wedge \neg\tau)$	$\dots$
$m+1.$	$\neg(\tau \wedge \neg\tau) \to \neg\varphi_n(x_n; y)$	m; SL
$m+2.$	$\neg(\tau \wedge \neg\tau) \to \forall y \neg\varphi_n(x_n; y)$	$m+1$; QR
$m+3.$	$\forall y \neg\varphi_n(x_n; y) \to \neg\varphi_n$	QAx
$m+4.$	$\neg(\tau \wedge \neg\tau) \to \neg\varphi_n$	$m+2$,$m+3$; SL
$m+5.$	$\neg(\tau_0 \wedge \neg\tau) \to \forall x_n \neg\varphi_n$	$n+4$; QR
$m+6.$	$\neg\forall x_n \neg\varphi_n \to (\tau \wedge \neg\tau)$	$m+5$; SL
	$[\exists x_n \varphi_n \to (\tau \wedge \neg\tau)]$	

This shows that $\Sigma'_n \vdash_{\mathcal{L}^*} (\exists x_n \varphi_n \to (\tau \wedge \neg\tau))$. But $\Sigma'_n = \Sigma_n \cup \{\exists x_n \varphi_n\}$, so it follows that $\Sigma'_n \vdash_{\mathcal{L}^*} (\tau \wedge \neg\tau)$. By SL, we get the contradiction that Σ'_n is inconsistent.

Suppose that Σ^* is inconsistent. Let $\mathbf{D}$ be a deduction of $\tau \wedge \neg\tau$ from Σ^*. Only finitely many members of Σ^* are lines of $\mathbf{D}$. Any finite subset of Σ^* is a subset of some Σ_n. This gives us the contradiction that some Σ_n is inconsistent. Hence Γ^* has property (2).

Because either σ_n or $\neg\sigma_n$ belongs to Σ_{n+1} for each n and each $\Sigma_{n+1} \subseteq \Sigma^*$, Σ^* has property (3).

If $\sigma_n \in \Sigma^*$, then $\neg\sigma_n \notin \Sigma_{n+1}$ and so $\sigma_n \in \Sigma_{n+1}$. But this implies that $\varphi_n(x_n; c_{i_n}) \in \Sigma_{n+1} \subseteq \Sigma^*$ if σ_n is $\exists x_n \varphi_n$. Since every sentences of $\mathcal{L}^*$ is σ_n for some n, Σ^* has property (4). $\square$

The following fact will be useful in proving Lemma 4.5, the second main part of the proof of Completeness.

Lemma 4.4. *Let φ be a formula, let $x_1, \dots, x_n$ be variables, and let $t_1, \dots, t_n$ and $t'_1, \dots t'_n$ be terms without variables.*

$$\vdash (t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n) \to (\varphi(x_1; t_1) \dots (x_n; t_n) \to \varphi(x_1; t'_1), \dots, (x_n; t'_n)).$$

Here, e.g., $\varphi(x_1; t_1) \dots (x_n; t_n)$ is the result of replacing, for each i, the free occurrences of x_i in φ by occurrences of t_i.

Proof. By the Deduction theorem, it will be enough to show

$$\{t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n\} \vdash (\varphi(x_1; t_1) \ldots (x_n; t_n) \rightarrow \varphi(x_1; t'_1), \ldots, (x_n; t'_n)).$$

1.	$t_1 = t'_1$	Premise
..	$\cdots$	$\cdots$
..	$\cdots$	$\cdots$
..	$\cdots$	$\cdots$
n.	$t_n = t'_n$	Premise
$n+1$.	$t_1 = t'_1 \rightarrow (\varphi(x_1; t_1)(x_2; t_2) \cdots (x_n; t_n)$ $\rightarrow (\varphi(x_1; t'_1)(x_2; t_2) \cdots (x_n; t'_n))$	IdAx(b)
..	$\cdots$	$\cdots$
..	$\cdots$	$\cdots$
..	$\cdots$	$\cdots$
$2n$.	$t_n = t'_n \rightarrow (\varphi(x_1; t'_1) \cdots (x_{n-1}; t'_{n-1})(x_n; t_n)$ $\rightarrow (\varphi(x_1; t'_1) \cdots (x_{n-1}; t'_{n-1})(x_n; t'_n))$	IdAx(b)
$2n+1$.	$\varphi(x_1; t_1) \cdots (x_n; t_n) \rightarrow \varphi(x_1; t'_1) \cdots (x_n; t'_n)$	SL; 1,. . . ,2n □

Remark. The condition that the terms t_i and t'_i are without variables could be dropped and replaced by the obvious generalization of the restriction on the Identity Axiom Schema (b). (If we did this, we would have to drop the use of the Deduction Theorem, but that use was inessential.)

Lemma 4.5. *Let Σ^* be a set of sentences of a language $\mathcal{L}^*$ having properties (2), (3), and (4) described in the statement of Lemma 4.3. Then Σ^* is satisfiable.*

Proof. We first show that Σ^* is *deductively closed*: for any sentence σ of $\mathcal{L}^*$, if $\Sigma^* \vdash \sigma$ then $\sigma \in \Sigma^*$. To show this, assume that $\Sigma^* \vdash \sigma$. If also $\neg\sigma \in \Sigma^*$, then Σ^* is inconsistent, contradicting (2). By (3), $\sigma \in \Sigma^*$.

The combination of deductive closure and Lemma 4.4 has a consequence that we will use later in the proof.

(#) *Assume that $t_1, \ldots, t_n$ and $t'_1, \ldots, t'_n$ are terms with out variables and that $\Sigma^* \vdash t_i = t'_i$ for $1 \leq i \leq n$. Let τ be a sentence, and let τ' be the result of replacing, for $1 \leq i \leq n$, each occurrence of t_i in τ by an occurrence of t'_i. If $\tau \in \Sigma^*$ then $\tau' \in \Sigma^*$.*

Proof of (#). Assume that $\tau \in \Sigma^*$. Let $v_1, \ldots, v_n$ be distinct variables not occurring in τ. Applying Lemma 4.4 to the formula φ resulting from replacing, for $1 \leq i \leq n$, each occurrence of t_i in τ by an occurrence of v_i, we get that

$$\vdash (t_1 = t'_1 \wedge \cdots t_n = t'_n) \rightarrow (\tau \rightarrow \tau').$$

By SL, $\Sigma^* \vdash \tau'$. By deductive closure, $\tau' \in \Sigma^*$. □

We will define a model $\mathfrak{A}$ and prove that Σ^* is true in it. Every member of A will be denoted by a constant. If c_1 and c_2 are constants and the sentence $c_1 = c_2$ belongs to Σ^*, then c_1 and c_2 will have to denote the same constant. This is the motivation for the following.

Let C^* be the set of all constants of $\mathcal{L}^*$. Let $\sim$ be the relation on C^* defined by

$$c_1 \sim c_2 \;\Leftrightarrow\; c_1 = c_2 \in \Sigma^*.$$

We will prove that $\sim$ is an *equivalence relation on* C^*: that $\sim$ is reflexive, symmetric, and transitive.

For reflexivity, we must show that $c = c$ belongs to Σ^* for all members c of C^*. Since $c = c$ is an instance of Identity Axiom Schema (a), $\vdash c = c$ and so $\Sigma^* \vdash c = c$. By deductive closure, $c = c \in \Sigma^*$.

For symmetry, we must show that, for all members c_1 and c_2 of Σ^*, if $c_1 = c_2 \in \Sigma^*$ then $c_2 = c_1 \in \Sigma^*$. Assume that $c_1 = c_2 \in \Sigma^*$. By Exercise 3.6, $\Sigma^* \vdash c_2 = c_1$. By deductive closure, $c_2 = c_1 \in \Sigma^*$.

Before proving transitivity, we show that

$$\{c_1 = c_2,\, c_2 = c_3\} \;\vdash\; c_1 = c_3$$

for any constants c_1, c_2, and c_3.

1.	$c_1 = c_2$	Premise
2.	$c_2 = c_3$	Premise
3.	$c_2 = c_1$	1; Exercise 3.6
4.	$c_2 = c_1 \rightarrow (c_2 = c_3 \rightarrow c_1 = c_3)$	IdAx(b)
5.	$c_1 = c_3$	2,3,4; SL

For transitivity, we must show that, for all members c_1, c_2, and c_3 of Σ^*, if $c_1 = c_2 \in \Sigma^*$ and $c_2 = c_3 \in \Sigma^*$, then $c_1 = c_3 \in \Sigma^*$. Assume that $c_1 = c_2 \in \Sigma^*$ and $c_2 = c_3 \in \Sigma^*$. By what we have just proved, $\Sigma^* \vdash c_1 = c_3$. By deductive closure, $c_1 = c_3 \in \Sigma^*$.

For each $c \in \mathrm{C}^*$, let $[c]$ be the *equivalence class of c with respect to* $\sim$:

$$[c] = \{c' \mid c \sim c'\}.$$

The model $\mathfrak{A}$. We define a model $\mathfrak{A}$ for $\mathcal{L}^*$ as follows.

(i) $A = \{[c] \mid c \in \mathrm{C}^*\}$.

(ii) $p_{\mathfrak{A}} = \mathbf{T} \Leftrightarrow p \in \Sigma^*$, for each sentence symbol p.

(iii) For $n \geq 1$, $P_{\mathfrak{A}} = \{([c_1], \ldots, [c_n]) \mid Pc_1 \ldots c_n \in \Sigma^*\}$, for each n-place predicate symbol P.

(iv) $c_{\mathfrak{A}} = [c]$ for each $c \in \mathrm{C}^*$.

(v) For $n \geq 1$, $f_{\mathfrak{A}}([c_1], \ldots, [c_n]) = [c] \Leftrightarrow fc_1 \ldots c_n = c \in \Sigma^*$, for each n-place function symbol f.

We must show that the definitions given in clauses (iii) and (v) do not depend on the choice of elements of equivalence classes. In the case of clause (v), we need to show something additional. (See below.)

Here is the proof for clause (iii). Assume that $[c_i] = [c_i']$ for $1 \leq i \leq n$. By the definition of the equivalence classes, we have that $c_i \sim c_i'$ for $1 \leq i \leq n$. By the definition of $\sim$, we get that the sentence $c_i = c_i'$ belongs to Σ^* for $1 \leq i \leq n$. By two applications of (#), we get that $Pc_1 \ldots c_n \in \Sigma^*$ if and only if $Pc_1' \ldots c_n' \in \Sigma^*$.

A special case of the proof that clause (v) is independent of the choice of elements of equivalence classes is Exercise 4.7, and the proof for the general case is just like the proof for the special case.

The additional fact we need to show concerning clause (v) is that, for all f and all $c_1, \ldots c_n \in \mathrm{C}^*$, there is a $c \in \mathrm{C}^*$ such that

$$fc_1 \ldots c_n = c \in \Sigma^*.$$

Suppose there is no such c. By property (3) of Σ^*,

$$fc_1 \ldots c_n \neq c \in \Sigma^*$$

for all $c \in \mathrm{C}^*$. By property (4) of Σ^*,

$$\forall v_1 fc_1 \ldots c_n \neq v_1 \in \Sigma^*.$$

Since

$$\forall v_1 fc_1 \ldots c_n \neq v_1 \rightarrow fc_1 \ldots c_n \neq fc_1 \ldots c_n$$

is an instance of the Quantifier Axiom Schema,

$$\Sigma^* \vdash fc_1 \ldots c_n \neq fc_1 \ldots c_n.$$

But $fc_1 \ldots c_n = fc_1 \ldots c_n$ is an instance of Identity Axiom Schema (a), and so Σ^* is inconsistent, contradicting property (2) of Σ^*.

Let P be the property of being a sentence σ such that

$$\mathrm{tv}_{\mathfrak{A}}(\sigma) = \mathbf{T} \;\Leftrightarrow\; \sigma \in \Sigma^* .$$

We prove by induction on length that every sentence has property P.

Before we begin the proof, we need to prove a fact about terms. Say that a term t containing no variables has property Q if and only if, for every $c \in \mathrm{C}^*$,

$$\text{if } \mathrm{den}_{\mathfrak{A}}(t) = [c] \text{ then } c = t \in \Sigma^*,$$

where $\mathrm{den}_{\mathfrak{A}}(t)$ is the common value of the $\mathrm{den}^s_{\mathfrak{A}}(t)$. We prove by induction on length that all terms without variables have Q.

(1) If t is a constant, then $\mathrm{den}_{\mathfrak{A}}(t) = t_{\mathfrak{A}} = [t]$. By definition of $[c]$, $c = t$ belongs to Σ^* if and only if $[t] = [c]$. Thus t has Q.

(2) Assume that t is $ft_1 \ldots t_n$. Let $\mathrm{den}_{\mathfrak{A}}(t_i) = [c_i]$ for $1 \leq i \leq n$. All the t_i are shorter than t and so have Q. Hence the sentence $c_i = t_i$ belongs to Σ^* for each i. Let $\mathrm{den}_{\mathfrak{A}}(t) = [c]$. By the definition of $\mathrm{den}_{\mathfrak{A}}$, it follows that

$$\begin{aligned}
\mathrm{den}_{\mathfrak{A}}(fc_1 \ldots c_n) &= f_{\mathfrak{A}}([c_1], \ldots, [c_n]) \\
&= \mathrm{den}_{\mathfrak{A}}(ft_1 \ldots t_n) \\
&= \mathrm{den}_{\mathfrak{A}}(t) \\
&= [c].
\end{aligned}$$

By the definition of $f_{\mathfrak{A}}([c_1], \ldots, [c_n])$, we have that $fc_1 \ldots c_n = c$ belongs to Σ^*. By (#), $ft_1 \ldots t_n = c \in \Sigma^*$, i.e. $t = c \in \Sigma^*$.

Now we begin the inductive proof that every sentence has property P.

Case (1)(a): σ is a sentence symbol p. By clause (ii) of the definition of $\mathfrak{A}$, $p_{\mathfrak{A}} = \mathbf{T} \;\Leftrightarrow\; p \in \Sigma^*$.

Case (1)(b): σ is $Pt_1 \ldots t_n$ for some n-place predicate symbol P and some terms $t_1, \ldots, t_n$. Let $\mathrm{den}_{\mathfrak{A}}(t_i) = [c_i]$ for $1 \leq i \leq n$. Since the t_i have property Q, $c_i = t_i \in \Sigma^*$ for each i.

$$\begin{aligned}
\mathrm{tv}(Pt_1 \ldots t_n) = \mathbf{T} &\Leftrightarrow (\mathrm{den}_{\mathfrak{A}}(t_1), \ldots, \mathrm{den}_{\mathfrak{A}}(t_n)) \in P_{\mathfrak{A}} \\
&\Leftrightarrow ([c_1], \ldots, [c_n])) \in P_{\mathfrak{A}} \\
&\Leftrightarrow Pc_1 \ldots c_n \in \Sigma^* \\
&\Leftrightarrow Pt_1 \ldots t_n \in \Sigma^*,
\end{aligned}$$

where the last $\Leftrightarrow$ comes by **(#)**.

Case (1)(c): σ is $t_1 = t_2$ for some terms t_1 and t_2. The proof is similar to the proof of Case (1)(b), and we omit it.

Case (2): σ is $\neg\tau$ for some sentence τ. We want to show that $\mathrm{tv}_{\mathfrak{A}}(\neg\tau) = \mathbf{T}$ if and only if $\neg\tau \in \Sigma^*$. Consider the following biconditionals.

$$\begin{aligned} \mathrm{tv}_{\mathfrak{A}}(\neg\tau) = \mathbf{T} &\Leftrightarrow \mathrm{tv}_{\mathfrak{A}}(\tau) = \mathbf{F} \\ &\Leftrightarrow \tau \notin \Sigma^* \\ &\Leftrightarrow \neg\tau \in \Sigma^* . \end{aligned}$$

These biconditionals imply that $\mathrm{tv}_{\mathfrak{A}}(\neg\tau) = \mathbf{T}$ if and only if $\neg\tau \in \Sigma^*$.

The first biconditional is true by definition of $\mathrm{tv}_{\mathfrak{A}}$. The second biconditional is true because τ is shorter than σ and so has property P. To finish Case (2), we need only prove the third biconditional.

For the "if" direction, assume that $\neg\tau \in \Sigma^*$. If $\tau \in \Sigma^*$, then Σ^* is inconsistent, so property (2) of Σ^* implies that $\tau \notin \Sigma^*$. For the "only if" direction, assume that $\tau \notin \Sigma^*$. By (3), $\neg\tau \in \Gamma^*$.

Case (3). σ is $(\rho \to \tau)$ for some sentences ρ and τ. We want to show that $\mathrm{tv}_{\mathfrak{A}}((\rho \to \tau)) = \mathbf{T}$ if and only if $(\rho \to \tau) \in \Sigma^*$. Consider the following biconditionals.

$$\begin{aligned} \mathrm{tv}_{\mathfrak{A}}(\rho \to \tau) = \mathbf{T} &\Leftrightarrow \text{if } \mathrm{tv}_{\mathfrak{A}}(\rho) = \mathbf{T} \text{ then } \mathrm{tv}_{\mathfrak{A}}(\tau) = \mathbf{T} \\ &\Leftrightarrow \text{if } \rho \in \Sigma^* \text{ then } \tau \in \Sigma^* \\ &\Leftrightarrow (\rho \to \tau) \in \Sigma^* . \end{aligned}$$

These biconditionals imply that

$$v^*((\rho \to \tau)) = \mathbf{T} \ \text{ if and only if } \ (\rho \to \tau) \in \Sigma^*.$$

The first biconditional is true by definition of $\mathrm{tv}_{\mathfrak{A}}$. The second biconditional is true because ρ and τ are shorter than $(\rho \to \tau)$, and so both have property P. To finish Case (3), we need only prove the third biconditional.

For the "if" direction, assume that $(\rho \to \tau) \in \Sigma^*$ and $\rho \in \Sigma^*$. By MP, $\Sigma^* \vdash \tau$ and so deductive closure implies that $\tau \in \Sigma^*$.

For the "only if" direction, assume that if $\rho \in \Gamma^*$ then $\tau \in \Sigma^*$. Either $\rho \in \Sigma^*$ or $\rho \notin \Sigma^*$. Assume first that $\rho \notin \Sigma^*$. By (3), $\neg\rho \in \Sigma^*$. By SL and deductive closure, $(\rho \to \tau) \in \Sigma^*$. Now assume that $\rho \in \Sigma^*$

By our assumption, $\tau \in \Sigma^*$. By SL and deductive closure, $(\rho \to \tau) \in \Sigma^*$.

Case (4): σ is $\forall x\varphi$ for some formula φ and some variable x. Note that no variable other than x can be free in φ.

$$\begin{aligned} \mathrm{tv}_{\mathfrak{A}}(\forall x\varphi) = \mathbf{T} &\Leftrightarrow \text{ for all } s,\ \mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T} \\ &\Leftrightarrow \text{ for all } c \in \mathrm{C}^*, \text{ for all } s \text{ with } s(x) = [c],\ \mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T} \\ &\Leftrightarrow \text{ for all } c \in \mathrm{C}^*,\ \mathrm{tv}^s_{\mathfrak{A}}(\varphi(x;c)) = \mathbf{T} \\ &\Leftrightarrow \text{ for all } c \in \mathrm{C}^*,\ \varphi(x;c) \in \Sigma^* \\ &\Leftrightarrow \forall x\varphi \in \Sigma^* \end{aligned}$$

These biconditionals imply that $\mathrm{tv}_{\mathfrak{A}}(\forall x\varphi) = \mathbf{T}$ if and only if $\forall x\varphi \in \Sigma^*$. The first biconditional is by the definition of $\mathrm{tv}_{\mathfrak{A}}(\sigma)$ and the fact that no variable besides x occurs free in φ. The second biconditional is by the fact that no variable besides x occurs free in φ and the fact that $A = \{[c] \mid c \in \mathrm{C}^*\}$. The third biconditional is by the fact that $c_{\mathfrak{A}} = [c]$ for each $c \in \mathrm{C}^*$. The fourth biconditional by the fact that the sentences $\varphi(x;c)$ are shorter than σ and so have property P.

To see that the "if" part of the last biconditional holds, assume that assume that $\forall x\varphi \in \Sigma^*$ and let $c \in \mathrm{C}^*$. Notice that the sentence

$$\forall x\varphi \to \varphi(x;c)$$

is an instance of the Quantifier Axiom Schema. Thus $\Sigma^* \vdash \varphi(x;c)$. By deductive closure, $\varphi(x;c) \in \Sigma^*$.

The "only if" part of the last biconditional holds by (4).

Our proof that all sentences of $\mathcal{L}^*$ have property P is complete. Since, in particular, $\mathrm{tv}_{\mathfrak{A}}(\sigma) = \mathbf{T}$ for every member σ of σ^*, we have shown that σ^* is satisfiable in $\mathcal{L}^*$. □

Theorem 4.6. *Let Σ be a consistent set of sentences of $\mathcal{L}$. Then Σ is satisfiable, i.e., true in a model for $\mathcal{L}$.*

Proof. Let Σ^* be given by Lemma 4.3. By Lemma 4.5, Σ^* is true in a model $\mathfrak{A}^*$ for $\mathcal{L}^*$. Since $\Sigma \subseteq \Sigma^*$, Σ is true in $\mathfrak{A}^*$. Let $\mathfrak{A}$ be the *reduct* to $\mathcal{L}$ of $\mathfrak{A}^*$, i.e., the model gotten by discarding functions $c_{\mathfrak{A}^*}$ for constants c that are not constants of $\mathcal{L}$. Then Σ is true in $\mathfrak{A}$. □

Theorem 4.7 (Completeness). *For each $\mathcal{L}$, our deductive system for $\mathcal{L}$ is complete.*

Proof. This follows from Exercise 4.3 and Theorem 4.6.

Theorem 4.8 (Compactness). *Let Σ be a set of sentences and let φ be a formula. If $\Sigma \models \varphi$, then there is a finite subset Δ of Σ such that $\Delta \models \varphi$.*

Proof. Assume that $\Sigma \models \varphi$. By Completeness, $\Sigma \vdash \varphi$. Let **D** be a deduction of φ from Σ. Let Δ be the set of sentences in Σ that are lines of **D**. Then Δ is finite and $\Delta \vdash \varphi$. By Soundness, $\Delta \models \varphi$. □

Exercise 4.5 (Compactness, Second Form). Use Theorem 4.8 to show that if every finite subset of a set of sentences is satisfiable then the whole set is satisfiable.

Hint. If Σ is not satisfiable, then $\Sigma \models (\tau \wedge \neg\tau)$ for any sentence τ.

Exercise 4.6. By the size of a model $\mathfrak{A}$, we mean the size of the domain A. Assume that Σ is a set of sentences and that there are arbitrarily large finite models in which Σ is true. Prove that Σ is true in some infinite model.

Hint. For each n, describe a sentence that is true in all and only those models that have size $\geq n$. This sentence should be in the language of identity, $\emptyset$. Use these sentences and Exercise 4.5.

Theorem 4.9 (Löwenheim-Skolem Theorem). *Every satisfiable set of sentences in a countable language with has a countable model, a model $\mathfrak{A}$ such that A is countable.*

Proof. The model $\mathfrak{A}$ constructed in the proof of Lemma 4.5 is countable, since C^* is countable.

Exercise 4.7. In the proof of Lemma 5.11, clause (v) of the definition of the model $\mathfrak{A}$ says that

$$f_{\mathfrak{A}}([c_1], \ldots, [c_n])) = [c] \quad \text{iff} \quad fc_1 \ldots c_n = c \in \Sigma^*.$$

Show, in the special case $n = 2$, that this definition does not depend on the choice of elements of equivalence classes. In other words, assume that

(1) $[c_1] = [c'_1]$ and $[c_2] = [c'_2]$;

(2) $fc_1c_2 = c \in \Sigma^*$ and $fc'_1c'_2 = c' \in \Sigma^*$,

and prove that

$$[c] = [c'].$$

Hint. Use **(#)**.

Exercise 4.8. Suppose that we had made $\wedge$ an additional official symbol in our languages, with the definition of $\mathrm{tv}^s_{\mathfrak{A}}$ augmented by the clause:

$$\mathrm{tv}^s_{\mathfrak{A}}(\varphi \wedge \psi) = \mathbf{T} \;\leftrightarrow\; (\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathbf{T} \text{ and } \mathrm{tv}^s_{\mathfrak{A}}(\psi) = \mathbf{T}).$$

In the proof of Lemma 4.5, there would have been an extra case in the proof that all formulas have property P. Give the proof for this extra case.

5 Peano Arithmetic and a Subtheory of it

Some minor changes:

(a) We add v_0 to our list of variables.

(b) We make $\{\mathbf{0}, \mathbf{S}, \boldsymbol{<}, \boldsymbol{+}, \boldsymbol{\cdot}\}$ the language $\mathcal{L}^{\mathrm{A}}$ of arithmetic.

The only reason for (a) is that the absent-minded author of the notes would probably use v_0 as a variable anyway, and the change prevents this from counting as a mistake.

The main reason for (b) is that we will often be talking simultaneously about the symbols of $\mathcal{L}^{\mathrm{A}}$ and about the number 0, the functions S, $+$, and $\cdot$, and the relation $\leq$. Writing the symbols in boldface will keep us from confusing them with the objects they standardly denote. The other change effected by (b) is to replace $\leq$ not by $\boldsymbol{\leq}$ but by $\boldsymbol{<}$.

Let $\mathfrak{N} = (\mathbb{N}, 0, S, <, +, \cdot)$. ($S$ is the successor function.) $\mathfrak{N}$ is the *standard model of arithmetic*. A central question for the rest of the course is whether $\mathfrak{N}$ is *axiomatizable*, whether there is a set Σ of sentences of $\mathcal{L}^{\mathrm{A}}$ with the following properties:

(i) For every sentence σ of $\mathcal{L}^{\mathrm{A}}$, $\Sigma \vdash \sigma \Leftrightarrow \sigma \in \mathrm{Th}(\mathfrak{N})$ $(= \{\tau \mid \mathrm{tv}_{\mathfrak{N}}(\tau) = \mathbf{T}\})$.

(ii) Σ is computable: there is an algorithm for deciding whether any given sentence of of $\mathcal{L}^{\mathrm{A}}$ is a member of Σ.

Requirement (ii) is a bit vague. We'll consider a precise version of it later.

Peano Arithmetic (PA) is the natural attempt to axiomatize $\mathfrak{N}$.

Axioms of PA.

(a) Universal closures of the following eight formulas (where use employ some obvious abbreviations, conventions, and extra parentheses):

(1) $\mathbf{0} \neq \mathbf{S}v_0$;

(2) $\mathbf{S}v_0 = \mathbf{S}v_1 \rightarrow v_0 = v_1$;

(3) $v_0 \boldsymbol{\not<} \mathbf{0}$;

(4) $v_0 \boldsymbol{<} \mathbf{S}v_1 \leftrightarrow v_0 \boldsymbol{\leq} v_1$;

(5) $v_0 \boldsymbol{+} \mathbf{0} = v_0$;

(6) $v_0{+}\mathbf{S}v_1 = \mathbf{S}(v_0{+}v_1)$;

(7) $v_0{\cdot}\mathbf{0} = \mathbf{0}$;

(8) $v_0{\cdot}\mathbf{S}v_1 = (v_0{\cdot}v_1){+}v_0$.

(b) The *Schema of Induction*, consisting of the universal closures of all formulas of the form:

$$(\varphi(x;\mathbf{0} \wedge \forall x(\varphi \to \varphi(x;Sx))) \to \forall x\varphi.$$

In presenting the axioms of PA, we have used the notion of universal closure. The *universal closure* of a formula φ is the sentence gotten by φ by preceding it with universal quantifiers for all variables occurring free in φ, in increasing order by subscripts. For example, the universal closure of $v_0{+}\mathbf{S}v_1 = (v_0{+}v_1)$ is $\forall v_0 \forall v_1\ v_0{+}\mathbf{S}v_1 = \mathbf{S}(v_0{+}v_1)$.

Note that our imprecise requirement (ii) on axiomatization is clearly satisfied by PA.

Non-Standard Models. Neither PA nor any other set of axioms—computable or not—can characterize the model $\mathfrak{N}$ up to isomorphism, as the following consequence of Compactness shows.

Theorem 5.1. *There is a model* $\mathfrak{A}$ *for* $\mathcal{L}^{\mathrm{A}}$ *such sentences* $\mathrm{Th}(\mathfrak{A}) = \mathrm{Th}(\mathfrak{N})$ *and such that there is an* $a \in A$ *with* $\mathrm{den}_{\mathfrak{A}}(\mathbf{S}^n\mathbf{0}) <_{\mathfrak{A}} a$ *for all* n. *(Here* $\mathbf{S}^n$ *is a string of* n $\mathbf{S}$*'s.)*

Proof. We get a language $\mathcal{L}$ by adding a new constant c to $\mathcal{L}^{\mathrm{A}}$. Let

$$\Sigma = \mathrm{Th}(\mathfrak{N}) \cup \{\mathbf{S}^n\mathbf{0} < c \mid 0 \leq n\}.$$

To show that every finite subset of Σ is satisfiable, let Δ be a finite subset of Σ. Let m be greater than every n such that $\mathbf{S}^n\mathbf{0} < c$ belongs to Δ. Make $\mathfrak{N}$ into a model $\mathfrak{B}$ for $\mathcal{L}$ by setting $c_{\mathfrak{B}} = m$. Δ is true in $\mathfrak{B}$.

By the second form of Compactness, Σ is satisfiable. Let $\mathfrak{A}^*$ be a model in which Σ is true. Let $\mathfrak{A}$ be the reduct of $\mathfrak{A}^*$ to $\mathcal{L}^{\mathrm{A}}$. Let $a = c_{\mathfrak{A}^*}$. □

We are going to study a particular *finitely axiomatizable* subtheory Q of PA.

Axioms of Q. The axioms of Q are Axioms (1)–(8) above.

Remarks. Often "Peano Arithmetic" is used to refer to a set of axioms in the language $\{\mathbf{0}, \mathbf{S}, +, \cdot\}$, namely, our Axioms (1), (2), (5)-(8), and the Schema of Induction. What is usually called "Q" is the finite set consisting of (1), (2), (5)-(8), and an additional axiom, the universal closure of $(v_0 = \mathbf{0} \vee \exists v_1(v_0 = \mathbf{S}v_1)$. Our version of PA is that of Herbert Enderton's *A Mathematical Introduction to Logic*, and our Q is Enderton's theory A with one axiom removed.

We will show, using coding based on *Gödel numbering*, that many truths about $\mathcal{L}^A$ and deduction from PA can be coded by sentences of $\mathcal{L}^A$ itself. We will see that many of these sentences can be proved from PA, or even from the weak theory Q. We will use our ability to prove facts about PA from PA or from Q to show that there are sentences of $\mathcal{L}^A$ that are neither provable or refutable from PA.

Remark. By Completeness and Soundness, $\vdash$ and $\models$ are equivalent. We will usually write $\models$, even when we are mainly thinking about provability.

Lemma 5.2. *For all k,*

$$\mathrm{Q} \models (x \mathbf{<} \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow (x = \mathbf{0} \vee \ldots \vee x = \mathbf{S}^k\mathbf{0}))\,.$$

Proof. We proceed by mathematical induction on k. By Axiom (4),

$$\mathrm{Q} \models (x \mathbf{<} \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow (x \mathbf{<} \mathbf{S}^k\mathbf{0} \vee x = \mathbf{S}^k\mathbf{0}))\,.$$

If $k = 0$, our conclusion follows by Axiom (3). If $k > 0$, it follows by induction. □

An abbreviation. For models $\mathfrak{A}$ and terms t, we use $t_{\mathfrak{A}}$ as an abbreviation for $\mathrm{den}_{\mathfrak{A}}(t)$.

Lemma 5.3. *If t is a term without variables and $k = t_{\mathfrak{N}}$, then*

$$\mathrm{Q} \models t = \mathbf{S}^k\mathbf{0}\,.$$

Proof. We prove the lemma by induction on the length of t. The case that t is $\mathbf{0}$ is immediate.

Assume that t is $\mathbf{S}u$ (for some term u without variables). By induction, $\mathrm{Q} \models u = \mathbf{S}^{u_{\mathfrak{N}}}\mathbf{0}$. Hence $\mathrm{Q} \models \mathbf{S}u = \mathbf{S}^{u_{\mathfrak{N}}+1}\mathbf{0}$.

Assume next that t is $u_1 + u_2$. Let $j_1 = (u_1)_{\mathfrak{N}}$ and let $j_2 = (u_2)_{\mathfrak{N}}$. By induction, $\mathrm{Q} \models u_1 = \mathbf{S}^{j_1}\mathbf{0}$ and $\mathrm{Q} \models u_2 = \mathbf{S}^{j_2}\mathbf{0}$. Axiom (5) and j_2 applications of Axiom (6) give that

$$\mathrm{Q} \models \mathbf{S}^{j_1}\mathbf{0} \mathbf{+} \mathbf{S}^{j_2}\mathbf{0} = \mathbf{S}^{j_1+j_2}\mathbf{0}\,.$$

Applications of Axioms (7) and (8) give that $\mathrm{Q} \models \mathbf{S}^{j_1}\mathbf{0} \boldsymbol{\cdot} \mathbf{S}^{j_2}\mathbf{0} = \mathbf{S}^{j_1 \cdot j_2}\mathbf{0}$, for any j_1 and j_2. This allows us to handle the case that t is $u_1 \boldsymbol{\cdot} u_2$. □

Notational Conventions. If φ is a formula and $x_1 \ldots, x_n$ are variables, then we write $\varphi(x_1, \ldots, x_n)$ to denote the formula φ and also to indicate that no variables besides $x_1, \ldots, x_n$ occur free in φ. We then may write $\varphi(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are terms, as an abbreviation for $\varphi(x_1; t_1) \cdots (x_n; t_n)$. If $\mathfrak{A}$ is a model, then we may write $\mathfrak{A}$ *satisfies* $\varphi[a_1, \ldots, a_n]$ to mean that $\mathrm{tv}^s_{\mathfrak{A}}(\varphi) = \mathrm{if}$ $s(x_i) = a_i$ for $1 \leq i \leq n$.

Representing relations. Let T be a theory (a set of sentences) in a language $\mathcal{L}$ containing $\mathbf{0}$ and $\mathbf{S}$. A formula $\varphi(v_1, \ldots, v_n)$ of $\mathcal{L}$ *represents* $R \subseteq \mathbb{N}^n$ *in* T if, for all elements $a_1, \ldots, a_n$ of $\mathbb{N}$,

$$\begin{aligned} R(a_1, \ldots, a_n) &\Rightarrow T \models \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0})\,; \\ \neg R(a_1, \ldots, a_n) &\Rightarrow T \models \neg\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0})\,. \end{aligned}$$

Here we write $R(a_1, \ldots, a_n)$ to mean that $(a_1, \ldots, a_n) \in R$.

If some formula represents R in T, then we say that R is *representable in* T.

Representability is related to definability. If $\mathfrak{A}$ is a model for $\mathcal{L}$ and $R \subseteq A^n$, then R is *definable in* $\mathfrak{A}$ if there is a formula $\varphi(v_1, \ldots, v_n)$ of $\mathcal{L}$ such that, for any members $a_1, \ldots, a_n$ of A,

$$R(a_1, \ldots, a_n) \Leftrightarrow \mathfrak{A} \text{ satisfies } \varphi[a_1, \ldots, a_n]\,.$$

For such a φ, we say that φ *defines* R *in* $\mathfrak{A}$. One relation between representability and definability is the following. Suppose that $\mathfrak{A}$ is a model of a theory T (a model is which T is true) in a language containing $\mathbf{0}$ and $\mathbf{S}$. Suppose also that $A = \mathbb{N}$, that $\mathbf{0}_{\mathfrak{A}} = 0$, and that $\mathbf{S}_{\mathfrak{A}} = S$. Then any formula that represents a relation in T also defines that relation in $\mathfrak{A}$. The converse is not in general true.

We will define representability of functions as well as of relations. A natural definition would be: "$\varphi(v_1, \ldots, v_{n+1})$ represents f in T if and only if φ represents the graph of f in T," where the *graph of f* is the $(n+1)$-place relation that holds of $(a_1, \ldots, a_{n+1})$ if and only if $f(a_1, \ldots, a_n) = a_{n+1}$. For technical reasons, we will define a stronger notion, though it will turn out that the two notions are equivalent for any T containing Axioms (1)–(4).

If $f : \mathbb{N}^n \to \mathbb{N}$ and T is a theory in a language containing $\mathbf{0}$ and $\mathbf{S}$, then a formula $\varphi(v_1, \ldots, v_{n+1})$ *represents f in T* if, for all $a_1, \ldots, a_n$,

$$T \models \forall v_{n+1}(\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, v_{n+1}) \leftrightarrow v_{n+1} = \mathbf{S}^{f(a_1,\ldots,a_n)}\mathbf{0}\,.$$

Say that f is *representable in T* if some formula represents f in T.

Note that if T contains Axioms (1) and (2) and φ represents f in T then φ represents the graph of f in T. We will say that T *proves $\varphi(v_1, \ldots, v_{n+1})$ functional* if

$$T \models \forall v_1 \cdots \forall v_n \exists v_{n+1} \forall v_{n+2}(\varphi(v_1, \ldots, v_n, v_{n+2}) \leftrightarrow v_{n+2} = v_{n+1})\,.$$

If T proves $\varphi(v_1, \ldots, v_{n+1})$ functional and φ represents the graph of f in T, then φ represents f in T. The converse does not hold in general.

Exercise 5.1. Show that, for every sentence σ of $\mathcal{L}^{\mathrm{A}}$ that is atomic or negation of atomic,

$$\mathrm{Q} \models \sigma \leftrightarrow \sigma \text{ is true in } \mathfrak{N}.$$

Exercise 5.2. A formula φ of $\mathcal{L}^{\mathrm{A}}$ is Δ_0 if φ belongs to the smallest set containing the atomic formulas and closed under negation, forming conditionals, and *bounded quantification*. Closure of Δ_0 under conditionals means that if φ and ψ are Δ_0 then so is $(\varphi \to \psi)$. Closure of Δ_0 under bounded quantification means that

$$\psi \text{ is } \Delta_0 \to \begin{cases} \forall x(x{<}t \to \psi) \text{ is } \Delta_0\,; \\ \forall x)(x{\leq}t \to \psi) \text{ is } \Delta_0\,, \end{cases}$$

for any term t not containing x. The Σ_1 formulas of $\mathcal{L}^{\mathrm{A}}$ are those of the form $\exists x_1 \cdots \exists x_n\, \psi$, where ψ is Δ_0.

(a) Prove that, for any Δ_0 sentence σ, $\mathrm{Q} \models \sigma \leftrightarrow \sigma$ is true in $\mathfrak{N}$.

(b) Prove that, for any Σ_1 sentence σ, $\mathrm{Q} \models \sigma \leftrightarrow \sigma$is true in $\mathfrak{N}$.

Hints for Exercises 5.1 *and* 5.2.

For every atomic sentence σ, there are terms t_1 and t_2 such that σ is either $t_1 = t_2$ or $t_1 < t_2$. In doing Exercise 5.1 for the case $t_1 = t_2$, use Lemma 5.3. For the case $t_1 < t_2$, use Lemma 5.3 and then use Lemma 5.2.

To do Exercise 5.2, use induction on length.

Primitive recursive functions. For $n \geq 0$, a function $f : \mathbb{N}^n \to \mathbb{N}$ is *primitive recursive* just in case (I)-(III) below require it to be. (I.e., the primitive recursive functions form the smallest set of functions containing the functions of (I) and closed under the operations of (II) and (III).)

(I) The following are primitive recursive.

(a) $S : \mathbb{N} \to \mathbb{N}$;
(b) $I_i^n : \mathbb{N}^n \to \mathbb{N}$, for $1 \leq i \leq n \in \mathbb{N}$, where $I_i^n(a_1, \ldots, a_n) = a_i$;
(c) All constant functions $f : \mathbb{N}^n \to \mathbb{N}$.

(II) (Composition) If $f : \mathbb{N}^m \to \mathbb{N}$ and $g_1, \ldots, g_m : \mathbb{N}^n \to \mathbb{N}$ are primitive recursive, then so is $h : \mathbb{N}^n \to \mathbb{N}$, where

$$h(a_1, \ldots, a_n) = f(g_1(a_1, \ldots, a_n), \ldots, g_m(a_1, \ldots, a_n)).$$

(III) (Primitive Recursion) If $f : \mathbb{N}^n \to \mathbb{N}$ and $g : \mathbb{N}^{n+2} \to \mathbb{N}$ are primitive recursive, then so is $h : \mathbb{N}^{n+1} \to \mathbb{N}$, where

$$\begin{aligned} h(a_1, \ldots, a_n, 0) &= f(a_1, \ldots, a_n); \\ h(a_1, \ldots, a_n, S(b)) &= g(a_1, \ldots, a_n, b, h(a_1, \ldots, a_n, b)). \end{aligned}$$

We allow functions of zero arguments (e.g., the f of (III)), all of which are primitive recursive by (I)(c).

Recursive functions. A function is called *recursive* or *computable* just in case it is required to be by (I)-(III), with "primitive recursive" replaced by "recursive," plus (IV) below.

(IV) (μ-Operator) If $g : \mathbb{N}^{n+1} \to \mathbb{N}$ is recursive and

$$(\forall a_1 \in \mathbb{N}) \cdots (\forall a_n \in \mathbb{N})(\exists b \in \mathbb{N})\, g(a_1, \ldots, a_n, b) = 0,$$

then $f : \mathbb{N}^n \to \mathbb{N}$ is recursive, where

$$f(a_1, \ldots, a_n) = \mu b \, g(a_1, \ldots, a_n, b) = 0 \,,$$

and where "μb" means "the least b."

Lemma 5.4. *The relations and functions representable in* Q *are closed under complement, intersection, union, and bounded quantification. Intersection and union we construe as operations acting on pairs of relations that are subsets of the same* $\mathbb{N}^n$. *Bounded quantification consists of the two operations* $(f, R) \mapsto R'$ *and* $(f, R) \mapsto R''$, *where*

$$R'(a_1, \ldots, a_n) \Leftrightarrow (\forall a_{n+1})(a_{n+1} < f(a_1, \ldots, a_n) \ \Rightarrow \ R(a_1, \ldots, a_{n+1}));$$
$$R''(a_1, \ldots, a_n) \Leftrightarrow (\exists a_{n+1})(a_{n+1} < f(a_1, \ldots, a_n) \,\&\, R(a_1, \ldots, a_{n+1})).$$

Proof. If φ represents R, then $\neg\varphi$ represents the complement of R; if φ and ψ represent R and R^* respectively, then $(\varphi \wedge \psi)$ represents $R \cap R^*$; if φ and ψ represent R and R^* respectively, then $(\varphi \vee \psi)$ represents $R \cup R^*$.

We do the R'' case for closure under bounded quantification. The R' case is similar. Let $\varphi(v_1, \ldots, v_{n+1})$ and $\psi(v_1, \ldots, v_{n+1})$ represent f and R respectively.

Let $\chi(v_1, \ldots, v_n)$ be, for some appropriate variable z,

$$\exists v_{n+1} \exists z(\varphi(v_1, \ldots, v_n, z) \ \wedge \ v_{n+1} \mathbf{<} z \ \wedge \ \psi(v_1, \ldots, v_n, v_{n+1})) \,.$$

To see that χ represents R'' in Q, fix numbers $a_1, \ldots, a_n$. Since φ represents f, we have that

$$\mathrm{Q} \ \models \ \forall z(\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z) \leftrightarrow z = \mathbf{S}^{f(a_1, \ldots, a_n)}\mathbf{0}) \,.$$

Thus $\chi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0})$ is equivalent in Q to

$$\exists v_{n+1}(v_{n+1} \mathbf{<} \mathbf{S}^{f(a_1, \ldots, a_n)}\mathbf{0} \ \wedge \ \psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, v_{n+1})) \,.$$

By Lemma 5.2, $\chi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0})$ is equivalent in Q to

$$\psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, \mathbf{0}) \ \vee \ldots \vee \ \psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, \mathbf{S}^{f(a_1, \ldots, a_n) - 1}\mathbf{0}) \,,$$

(or to, say, $\mathbf{0} \neq \mathbf{0}$ if $f(a_1, \ldots, a_n) = 0$). Since ψ represents R, this formula is provable or refutable in Q according to whether or not $R''(a_1, \ldots, a_n)$ holds. □

Lemma 5.5. *All the functions under clause (I) (in the definition of the primitive recursive functions) are representable in* Q.

Proof. They are represented by atomic formulas. For example, I_i^n is represented by $v_{n+1} = v_i$, because

$$\mathrm{Q} \models \forall v_{n+1}(v_{n+1} = \mathbf{S}^{a_i}\mathbf{0} \leftrightarrow v_{n+1} = \mathbf{S}^{I_i^n(\mathbf{S}^{a_1}\mathbf{0},\ldots,\mathbf{S}^{a_n}\mathbf{0})}),$$

since $I_i^n(a_1, \ldots, a_n) = a_i$. Indeed, $\emptyset \models$ the displayed sentence. □

Lemma 5.6. *The functions representable in* Q *are closed under composition (II).*

Proof. Given representable f and $g_1, \ldots, g_m$, as in the statement of (II), let $\psi_1(v_1, \ldots, v_{n+1}), \ldots, \psi_m(v_1, \ldots, v_{n+1})$ represent $g_1, \ldots, g_m$ respectively and let $\chi(v_1, \ldots, v_{m+1})$ represent f. Let $\varphi(v_1, \ldots, v_{n+1})$ be, for appropriate variables $x_1, \ldots, x_m$,

$$\exists x_1 \cdots \exists x_m(\psi_1(v_1, \ldots, v_n, x_1) \wedge \ldots \\ \wedge \psi_m(v_1, \ldots, v_n, x_m) \wedge \chi(x_1, \ldots, x_m, v_{n+1}))\,.$$

Let $a_1, \ldots, a_n \in \mathbb{N}$. For each j,

$$\mathrm{Q} \models \forall x_j \left(\psi_j(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, x_j) \leftrightarrow x_j = \mathbf{S}^{g_j(a_1,\ldots,a_n)}\mathbf{0}\right)\,.$$

Thus $\mathrm{Q} \models$

$$\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, v_{n+1}) \leftrightarrow \chi(\mathbf{S}^{g_1(a_1,\ldots,a_n)}\mathbf{0}, \ldots, \mathbf{S}^{g_m(a_1,\ldots a_n)}\mathbf{0}, v_{n+1})\,.$$

But $\mathrm{Q} \models$

$$\chi(\mathbf{S}^{g_1(a_1,\ldots,a_n)}\mathbf{0}, \ldots, \mathbf{S}^{g_m(a_1,\ldots,a_n)}\mathbf{0}, v_{n+1}) \\ \leftrightarrow v_{n+1} = \mathbf{S}^{f(g_1(a_1,\ldots,a_n),\ldots,g_m(a_1,\ldots a_n))}\mathbf{0}\,.$$

Therefore $\mathrm{Q} \models$

$$\forall v_{n+1}(\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, v_{n+1}) \leftrightarrow v_{n+1} = \mathbf{S}^{f(g_1(a_1,\ldots,a_n),\ldots,g_m(a_1,\ldots a_n))}\mathbf{0}\,.$$

□

Exercise 5.3. The part of the proof of Lemma 5.3 for the case that t is of the form $u_1 \cdot u_2$ is only hinted at in these notes. Give the proof for that case.

Exercise 5.4. Prove that addition, multiplication, and the factorial function are primitive recursive. The factorial function is defined by

$$f(n) = n! = \text{the product of the numbers } 1, \dots n,$$

for $n \geq 1$, and $0! = 1$.

Lemma 5.7. *A relation R is representable in* Q *if and only if its characteristic function K_R is representable in* Q*, where*

$$K_R(a_1, \dots, a_n) = \begin{cases} 1 & \text{if } R(a_1, \dots, a_n)\,; \\ 0 & \text{if } \neg R(a_1, \dots, a_n)\,. \end{cases}$$

Exercise 5.5. Prove Lemma 5.7.

Our next goal is to show that the functions representable in Q are closed under the μ-operator (IV). This would be easy if the sentence $\forall v_1 \forall v_2 (v_1 < v_2 \lor v_1 = v_2 \lor v_2 < v_1)$ were provable in Q. We could have made this sentence an axiom of a strengthening of Q, as does Enderton in the book cited earlier. But we did not do this, so our argument will be a little complicated.

Let $\mathbf{WC}(v_1)$ be the following formula:

$$(\mathbf{0} \boldsymbol{\leq} v_1 \ \land \ \forall v_2 (v_2 \boldsymbol{<} v_1 \rightarrow \mathbf{S} v_2 \boldsymbol{\leq} v_1))\,.$$

Think of $\mathbf{WC}$ as "weakly comparable."

Lemma 5.8. *For every natural number k,*

(a) Q $\models$ $\mathbf{WC}(\mathbf{S}^k\mathbf{0})$*;*

(b) Q $\models$ $\forall v_1 (\mathbf{WC}(v_1) \rightarrow (v_1 \boldsymbol{<} \mathbf{S}^k\mathbf{0} \lor v_1 = \mathbf{S}^k\mathbf{0} \lor \mathbf{S}^k\mathbf{0} \boldsymbol{<} v_1))$*.*

Proof. That Q $\models \mathbf{WC}(\mathbf{0})$ follows from Axiom (3). Fix $k > 0$. By Exercise 5.1 (or by Lemma 5.2), we know that Q $\models \mathbf{0} \boldsymbol{\leq} \mathbf{S}^k\mathbf{0}$. An application of Lemma 5.2 gives that

$$\mathrm{Q} \models v_2 \boldsymbol{<} \mathbf{S}^k\mathbf{0} \rightarrow (v_2 = \mathbf{0} \lor \dots \lor v_2 = \mathbf{S}^{k-1}\mathbf{0})\,.$$

But then

$$\mathrm{Q} \models v_2 \boldsymbol{<} \mathbf{S}^k\mathbf{0} \rightarrow (\mathbf{S} v_2 = \mathbf{S}^1\mathbf{0} \lor \dots \lor \mathbf{S} v_2 = \mathbf{S}^k\mathbf{0})\,.$$

(a) follows by Lemma 5.2.

We prove (b) by induction on k. The case $k = 0$ comes from the first conjunct of $\mathbf{WC}(v_1)$. For the induction step note that, by Axiom (4), $\mathrm{Q} \models (v_1 {\leq} \mathbf{S}^k\mathbf{0} \rightarrow v_1 {<} \mathbf{SS}^k\mathbf{0})$ and that, by the second conjunct of $\mathbf{WC}(v_1)$,

$$\mathrm{Q} \ \models \ (\mathbf{S}^k\mathbf{0}{<}v_1 \wedge \mathbf{WC}(v_1)) \ \rightarrow \ \mathbf{SS}^k\mathbf{0}{\leq}v_1 \,. \qquad \square$$

Lemma 5.9. *The functions representable in* Q *are closed under the* μ*-operator (IV).*

Proof. Suppose that $\varphi(v_1, \ldots, v_{n+2})$ represents g in Q and suppose that

$$(\forall a_1 \in \mathbb{N}) \cdots (\forall a_n \in \mathbb{N})(\exists b \in \mathbb{N})\, g(a_1, \ldots, a_n, b) = 0 \,.$$

Let f be given by

$$f(a_1, \ldots, a_n) = \mu b\, g(a_1, \ldots, a_n, b) = 0 \,.$$

Let $\psi(v_1, \ldots, v_{n+1})$ be, for an appropriate z,

$$\mathbf{WC}(v_{n+1}) \ \wedge \ \varphi(v_1, \ldots, v_{n+1}, \mathbf{0}) \ \wedge \ \forall z(z{<}v_{n+1} \rightarrow \neg\varphi(v_1, \ldots, v_n, z, \mathbf{0})) \,.$$

To see that ψ represents f in Q, fix $a_1, \ldots, a_n$. Using part (a) of Lemma 5.8 and the fact that φ represents g in Q, we deduce that

$$\mathrm{Q} \ \models \ \mathbf{WC}(\mathbf{S}^{f(a_1,\ldots,a_n)}\mathbf{0}) \ \wedge \ \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, \mathbf{S}^{f(a_1,\ldots,a_n)}\mathbf{0}, \mathbf{0}), .$$

Using the fact that φ represents g in Q and using Lemma 5.2, we get that

$$\mathrm{Q} \ \models \ \forall z(z{<}\mathbf{S}^{f(a_1\ldots,a_n)}\mathbf{0} \rightarrow \neg\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z, \mathbf{0})) \,.$$

Combining these two facts we get that

$$\mathrm{Q} \models \psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, \mathbf{S}^{f(a_1,\ldots,a_n)}\mathbf{0}) \,.$$

Moreover, the second of the two facts and part (b) of Lemma 5.8 give that

$$\mathrm{Q} \ \models \ (\forall z)((\mathbf{WC}(z) \ \wedge \ \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z, \mathbf{0})) \ \rightarrow \ \mathbf{S}^{f(a_1\ldots,a_n)}\mathbf{0}{\leq}z) \,.$$

Since $\mathbf{WC}(z)$ and $\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z, \mathbf{0})$ are conjuncts of the formula $\psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z)$,

$$\mathrm{Q} \models (\forall z)(\psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z) \to \mathbf{S}^{f(a_1 \ldots, a_n)}\mathbf{0} \leq z)\,.$$

Since $\mathrm{Q} \models \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, \mathbf{S}^{f(a_1, \ldots, a_n)}\mathbf{0}, \mathbf{0})$, consideration of the last conjunct of $\psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z)$ shows us that

$$\mathrm{Q} \models (\forall z)(\psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z) \to \mathbf{S}^{f(a_1, \ldots, a_n)}\mathbf{0} \not< z)\,.$$

Thus

$$\mathrm{Q} \models (\forall z)(\psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, z) \to z = \mathbf{S}^{f(a_1, \ldots, a_n)}\mathbf{0})\,. \qquad \square$$

Corollary 5.10. *A function is representable in* Q *if its graph is representable in* Q.

Proof. Let R be the graph of $f : \mathbb{N}^n \to \mathbb{N}$.

$$f(a_1, \ldots, a_n) = \mu b\, K_{\neg R}(a_1, \ldots, a_n, b) = 0\,. \qquad \square$$

Lemma 5.11. *The relation $<$ and the functions $+$ amd $\cdot$ are representable in Q.*

Proof. By Exercise 5.1, $<$ and the graphs of $+$ and $\cdot$, are represented by $v_1 < v_2$, $v_1 + v_2 = v_3$, and $v_1 \cdot v_2 = v_3$ respectively. Use Corollary 5.10 or the fact that every theory proves the last three formulas functional. $\square$

Exercise 5.6. Prove that if $f : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive so is $g : \mathbb{N}^2 \to \mathbb{N}$, where g is defined by $g(a, b) = f(b, a)$.

Lemma 5.12. $\{(a, b) \mid a$ divides $b\}$ *is representable in* Q.

Proof. For any a and b belonging to $\mathbb{N}$,

$$a \text{ divides } b \leftrightarrow (\exists c \leq b)\, a \cdot c = b\,. \qquad \square$$

Lemma 5.13. *(a) The set of all prime numbers is representable in* Q.

(b) The set of all pairs of adjacent primes is representable in Q, *where (a, b) is a pair of adjacent primes if and only if $a < b$, both a and b are prime, and there is no prime c such that $a < c < b$.*

Exercise 5.7. Prove Lemma 5.13.

Our next goal is to prove that exponentation is representable in Q. By exponentiation, we mean the function E defined by setting $E(a_1, a_2) = a_1^{a_2}$. We will use the following number-theoretic theorem.

Lemma 5.14 (Chinese Remainder Theorem). *Let the positive integers $d_0, \ldots, d_n$ be relatively prime. Let $a_i < d_i$ for each $i \leq n$. Then there is a c such that, for each $i \leq n$, a_i is the remainder when c is divided by d_i.*

Proof. For any $c \in \mathbb{N}$, let $\mathbf{F}(c) = (r_0, \ldots, r_n)$, where each r_i is the remainder when c is divided by d_i.

Suppose c_1 and c_2 are distinct numbers smaller than $\prod_{i \leq n} d_i$ ($= d_0 \cdot \ldots \cdot d_n$). If $\mathbf{F}(c_1) = \mathbf{F}(c_2)$, then each d_i divides $|c_1 - c_2|$ and so, since the d_i are relatively prime, $\prod_{i \leq n} d_i$ divides $|c_1 - c_2|$. This contradiction shows that $\mathbf{F}(c_1) \neq \mathbf{F}(c_2)$.

Thus $\mathbf{F}(c)$ takes on $\prod_{i \leq n} d_i$ distinct values for $c < \prod_{i \leq n} d_i$. But each of these values is of the form $(r_0, \ldots, r_n)$ with each $r_i < d_i$. There are only $\prod_{i \leq n} d_i$ such $(r_0, \ldots, r_n)$, so one of the $\mathbf{F}(c)$ must be $(a_0, \ldots, a_n)$. □

Lemma 5.15. *For any positive integer m, the numbers $1 + (i+1) \cdot m!$, $i \leq m$, are relatively prime.*

Proof. Let i and j be distinct numbers $\leq m$. Suppose that some prime p divides both $1 + (i + 1) \cdot m!$ and $1 + (j + 1) \cdot m!$, with i and $j \leq m$. Then p divides $|i - j| \cdot m!$. Since p cannot divide $m!$, it follows that p must divide $|i - j|$. But $|i - j| \leq m$, and thus we have the contradiction that p divides $m!$. □

For elements c, d, and i of $\mathbb{N}$, let $r(c, d, i)$ be the remainder when c is divided by $1 + (i + 1) \cdot d$.

Order the set of all pairs (a, b) of natural numbers first by $\max\{a, b\}$ and then lexicographically. For pairs (a, b), let $n(a, b)$ be the number of pairs preceding (a, b) in this ordering. Define $q_1 : \mathbb{N} \to \mathbb{N}$ and $q_2 : \mathbb{N} \to \mathbb{N}$ by setting $q_1(n(a, b)) = a$ and $q_2(n(a, b)) = b$.

Lemma 5.16. *The functions r, n, q_1, and q_2 are representable in* Q.

Exercise 5.8. Prove Lemma 5.8.

Hint. Find a relation R representable representable in Q such that $r(c,d,i) = \mu b R(c,d,i,b)$, and apply the trick used in the proof of Lemma 5.10. Next show that max is representable in Q. Next compute $n(a,b)$ using $\max\{a,b\}$, a, b, and $K_{\leq}(b,a)$ Finally, use closure under bounded quantification and the μ operator to compute q_1 and q_2.

Lemma 5.17. *For any natural numbers n and $a_0, \ldots, a_n$, there are c and d such that*

$$(\forall i \leq n)\, r(c,d,i) = a_i\,.$$

Proof. Given n and $a_0, \ldots a_n$, let $m = \max\{n, a_0, \ldots, a_n\}$. Let $d = m!$. Since the $1 + (i+1) \cdot d$ are relatively prime, let c be given by the Chinese Remainder Theorem. (Note that each $a_i < 1 + (i+1) \cdot d$.) □

Lemma 5.18. *Exponentiation is representable in* Q.

Proof. Define functions $f : \mathbb{N}^2 \to \mathbb{N}$ and $E^* : \mathbb{N}^2 \to \mathbb{N}$ by

$$\begin{aligned} f(m,i) &= r(q_1(m), q_2(m), i)\,; \\ E^*(a,b) &= \mu m\, (f(m,0) = 1 \wedge (\forall i \leq b)\, f(m,i+1) = f(m,i) \cdot a)\,. \end{aligned}$$

Both f and E^* are representable in Q. Moreover, we have that

$$(\forall a \in \mathbb{N})(\forall b \in \mathbb{N})(\forall i \leq b)\, f(E^*(a,b), i) = a^i\,.$$

Thus $a^b = f(E^*(a,b), b)$ for all a and b. □

Lemma 5.19. *The function $a \mapsto p_a$ is representable in* Q, *where p_a is the $a+1$st prime.*

Proof. We shall show that, for any a and b belonging to $\mathbb{N}$, $p_a = b$ if and only if b is prime and there is a $c \leq b^{a^2}$ such that

(i) 2 does not divide c;

(ii) For all $q < b$ and all $r \leq b$, if (q,r) is a pair of adjacent primes, then

$$(\forall j < c)(q^j \text{ divides } c \leftrightarrow r^{j+1} \text{ divides } c)\,.$$

(iii) b^a divides c and b^{a+1} does not.

To see this, fix a and b and first note that if $p_a = b$ and

$$c = {p_0}^0 \cdot {p_1}^1 \cdot \ldots \cdot {p_a}^a ,$$

then $c \leq b^{a^2}$ and c satisfies (i)–(iii).

Suppose that b is prime and that c satisfies (i)–(iii).

By induction we show that

$$(\forall i \in \mathbb{N})(p_i \leq b \rightarrow ({p_i}^i \text{ divides } c \wedge {p_i}^{i+1} \text{ does not divide } c)) .$$

For $i = 0$ this is given by (i). Suppose that $i = j + 1$ and that ${p_j}^j$ divides c but ${p_j}^{j+1}$ does not. The desired conclusion follows from (ii) with $q = p_j$ and $r = p_i$, since $j < {p_j}^j \leq c$.

Now b is prime, and so $b = p_j$ for some j. Thus b^j divides c and b^{j+1} does not. By (iii), it follows that $j = a$. □

For natural numbers $a_0, \ldots, a_m$, let

$$\langle a_0, \ldots, a_m \rangle = {p_0}^{a_0+1} \cdot \ldots \cdot {p_m}^{a_m+1} .$$

For $m = -1$, let $\langle\ \rangle = 1$. Let Seq be the set of all a such that $a = \langle a_0, \ldots, a_m \rangle$ for some $m \geq -1$ and some $a_0, \ldots, a_m$. For elements a and b of $\mathbb{N}$, let

$$(a)_b = \mu n \, ({p_b}^{n+2} \text{ does not divide } a) .$$

Lemma 5.20. *(a) For each $m \in \mathbb{N}$, the function*

$$(a_0, \ldots, a_{m-1}) \mapsto \langle a_0, \ldots, a_{m-1} \rangle$$

is representable in Q*. (b) The function $(a, b) \mapsto (a)_b$ is representable in* Q*. (c)* Seq *is representable in* Q*.*

Proof. (a) holds by closure under composition. For (b), apply the μ-operator to the characteristic function of the relation

$${p_b}^{n+2} \text{ divides } a .$$

For (c), note that

$$a \in \text{Seq} \leftrightarrow a > 0 \wedge (\forall i \leq a)(p_{i+1} \text{ divides } a \rightarrow p_i \text{ divides } a) . \qquad \square$$

For $a \in \mathbb{N}$, let

$$\mathrm{lh}(a) = \mu n\,(a = 0 \vee p_n \text{ does not divide } a)\,.$$

For a and b elements of $\mathbb{N}$, let

$$a\lceil b = \mu n\,(a = 0 \vee (n \neq 0 \wedge (\forall j{<}b)(\forall k{<}a)({p_j}^k \text{ divides } a \rightarrow {p_j}^k \text{ divides } n)))\,.$$

The following lemma follows easily from the definitions and earlier results.

Lemma 5.21. *The functions* lh *and* $(a, b) \mapsto (a\lceil b)$ *are representable in* Q. *For all* $m \geq -1$ *and all* $a_0, \ldots, a_m$*,*

(i) $\mathrm{lh}(\langle a_0, \ldots, a_m\rangle) = m + 1$ *;*

(ii) $\langle a_0, \ldots, a_m\rangle\lceil b = \langle a_0, \ldots, a_{b-1}\rangle$ if $b \leq m + 1$ *.*

For $n \in \mathbb{N}$ and $h : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$, let $\bar{h} : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ be given by

$$\bar{h}(a_1, \ldots, a_n, b) = \langle h(a_1, \ldots, a_n, 0), \ldots, h(a_1, \ldots, a_n, b - 1)\rangle\,.$$

Lemma 5.22. *The set of functions representable in* Q *is closed under primitive recursion* (III)*.*

Proof. Let $h : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ be defined from $f : {}^{n}\mathbb{N} \rightarrow \mathbb{N}$ and $g : {}^{n+2}\mathbb{N} \rightarrow \mathbb{N}$ as in the statement of (III). Assume that f and g are representable in Q. We first show that $\bar{h}$ is representable:

$$\begin{aligned}\bar{h}(a_1, \ldots, a_n, b) \;=\; & \mu m\,(m \in \mathrm{Seq} \wedge \mathrm{lh}(m) = b \wedge \\ & \quad (\forall i < b)((i = 0 \wedge (m)_i = f(a_1, \ldots, a_n)) \vee \\ & \quad\; (\exists j < i)(i = j + 1 \wedge (m)_i = g(a_1, \ldots, a_n, j, (m)_j))))\,.\end{aligned}$$

Now we note that

$$h(a_1, \ldots, a_n, b) = (\bar{h}(a_1, \ldots, a_n, b + 1))_b\,. \qquad \square$$

Theorem 5.23. *Every recursive function is representable in* Q.

Proof. This follows from Lemmas 5.5, 5.6, 5.22, and 5.9. $\square$

6 Incompleteness

Our next goal is to show that various functions coding syntactical relations in languages such as $\mathcal{L}^{\mathrm{A}}$ are primitive recursive.

Lemma 6.1. *If* $t(v_1, \ldots, v_n)$ *is a term of* $\mathcal{L}^{\mathrm{A}}$*, then the function*

$$(a_1, \ldots, a_n) \mapsto (t(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}))_{\mathfrak{N}}$$

is primitive recursive.

Proof. Successor and the constant function with value 0 are primitive recursive by (I). Addition and multiplication are primitive recursive by Exercise 5.4. Exponentiation is defined by primitive recursion, using multiplication and other functions we know to be primitive recursive. The primitive recursiveness of functions given by general terms can be proved by induction on length, using composition and the I^n_i. □

Lemma 6.2. *The functions* sg, pred, *and* $\dot{-}$ *are primitive recursive, where*

$$\begin{aligned} \mathrm{sg}(a) &= \begin{cases} 1 & \text{if } a > 0\,; \\ 0 & \text{if } a = 0\,; \end{cases} \\ \mathrm{pred}(a) &= \begin{cases} a-1 & \text{if } a > 0\,; \\ 0 & \text{if } a = 0\,; \end{cases} \\ a \dot{-} b &= \begin{cases} a-b & \text{if } a \geq b\,; \\ 0 & \text{if } a < b\,; \end{cases} \end{aligned}$$

Exercise 6.1. Prove Lemma 6.2.

Hint. Use primitive recursion.

Call a relation *primitive recursive* or *recursive* if its characteristic function is primitive recursive or recursive.

Lemma 6.3. *The set of all primitive recursive relations is closed under complement, intersection, and union. The relation* $<$ *is primitive recursive.*

Proof. Note that $K_{\neg R}(a_1, \ldots, a_n) = 1 \dot{-} K_R(a_1, \ldots a_n)$, that $K_{R \cap S}(a_1, \ldots, a_n) = K_R(a_1, \ldots, a_n) \cdot K_S(a_1, \ldots, a_n)$, that $K_{R \cup S}(a_1, \ldots, a_n) = \mathrm{sg}(K_R(a_1, \ldots, a_n) + K_S(a_1, \ldots, a_n))$, and that $K_<(a, b) = \mathrm{sg}(b \dot{-} a)$. □

Lemma 6.4. *The set of primitive recursive functions is closed under the two operations* $f \mapsto g$ *given by*

$$
\begin{aligned}
g(a_1, \dots, a_n, b) &= \sum_{b' < b} f(a_1, \dots, a_n, b')\,; \\
g(a_1, \dots, a_n, b) &= \prod_{b' < b} f(a_1, \dots, a_n, b')\,.
\end{aligned}
$$

(We consider the empty product to have value 1*.)*

Proof. We consider only the case of $\sum$. That of $\prod$ is similar. We have

$$
\begin{aligned}
g(a_1, \dots, a_n, 0) &= 0\,; \\
g(a_1, \dots, a_n, S(b)) &= g(a_1, \dots, a_n, b) + f(a_1, \dots, a_n, b)\,.
\end{aligned}
$$

Thus g comes by primitive recursion from functions that are primitive recursive if f is. $\square$

Lemma 6.5. *The set of primitive recursive relations and functions is closed under bounded quantification.*

Proof. Let $R'(a_1, \dots, a_n) \Leftrightarrow (\exists b < f(a_1, \dots, a_n))\, R(a_1, \dots, a_n, b)$. Then

$$
K_{R'}(a_1, \dots, a_n) = \mathrm{sg}\left(\sum_{b < f(a_1, \dots, a_n)} K_R(a_1, \dots, a_n, b) \right). \qquad \square
$$

Lemma 6.6. *The set of primitive recursive functions is closed under the* bounded μ-operator*, i.e., under* $(f, g) \mapsto h$*, where*

$$
h(a_1, \dots, a_n) = \mu b\, (b = f(a_1, \dots, a_n) \lor g(a_1, \dots, a_n, b) = 0)\,.
$$

Exercise 6.2. Prove Lemma 6.6.

Lemma 6.7. *The relations and functions representable in* Q *by Lemmas 5.12, 5.13, 5.19, 5.20, and 5.21 are primitive recursive.*

Proof. Except for the case of Lemma 5.19, the proofs of representability, with minor modifications, yield proofs of primitive recursiveness. The main thing to note is that the uses of the μ-operator in defining $(a)_b$, $a\lceil b$, and $\mathrm{lh}(a)$, are equivalent to the corresponding uses of the bounded μ-operator, with the bound function f in each case a constant function with value a.

For Lemma 5.19, Euclid's proof that there are infinitely many primes shows that

$$p_{S(a)} = \mu b \left(b \leq 1 + \prod_{i<a} p_i \, \wedge \, p_a < b \, \wedge \, b \text{ is prime} \right)$$

for each $n \in \mathbb{N}$. Using this fact, we can define $a \mapsto \prod_{i<a} p_i$ by primitive recursion from functions we can show to be primitive recursive. Using the fact again, we get that $a \mapsto p_a$ is primitive recursive. □

Exercise 6.3. Explain why the definitions of lh and $(a, b) \mapsto (a\lceil b)$ yield lemma 5.21; i.e., explain how the uses the μ operator in these definitions are bounded.

Define $* : \mathbb{N}^2 \to \mathbb{N}$ by

$$a * b = a \cdot \prod_{i<\mathrm{lh}(b)} p_{\mathrm{lh}(a)+i}{}^{(b)_i+1} .$$

The following lemma is evident.

Lemma 6.8. *The function $*$ is primitive recursive. For m and $n \geq -1$ and for any elments $a_0, \ldots, a_m$, $b_0, \ldots, b_n$ of $\mathbb{N}$,*

$$\langle a_0, \ldots, a_m \rangle * \langle b_0, \ldots, b_n \rangle = \langle a_0, \ldots, a_m, b_0, \ldots, b_n \rangle .$$

For any $n \in \mathbb{N}$ and any $f : \mathbb{N}^{n+1} \to \mathbb{N}$, define a function $(a_1, \ldots, a_n, b) \mapsto *_{i<b} f(a_1, \ldots, a_n, i)$ by

$$\begin{aligned} *_{i<0} f(a_1, \ldots, a_n, i) &= 1 ; \\ *_{i<b+1} f(a_1, \ldots, a_n, i) &= (*_{i<b} f(a_1, \ldots, a_n, i)) * f(a_1, \ldots, a_n, b) . \end{aligned}$$

The following lemma is also evident.

Lemma 6.9. *The function $(a_1, \ldots, a_n, b) \mapsto *_{i<b} f(a_1, \ldots, a_n, i)$ is primitive recursive if f is primitive recursive.*

We next assign *symbol numbers* to all the symbols of $\mathcal{L}^{\mathrm{A}}$. To each variable v_i, we assign the symbol number $2i$. The symbol numbers of the remaining symbols are given as follows.

$\neg$	1	$\mathbf{0}$	13
$\rightarrow$	3	$\mathbf{S}$	15
(	5	$+$	17
)	7	$\cdot$	19
$=$	9	$\mathbf{E}$	21
$\forall$	11	$<$	23

We want what we say to apply to other languages. Fix a language $\mathcal{L}$. Assume that symbol numbers have been assigned to the non-logical symbols of $\mathcal{L}$ so that the following relations are primitive recursive:

$$\{(k,m) \mid k \text{ is the symbol number of an } m\text{-place relation symbol}\}\,;$$
$$\{(k,m) \mid k \text{ is the symbol number of an } m\text{-place function symbol}\}\,.$$

Note that this is true for $\mathcal{L}^{\mathrm{A}}$.

We next assign numbers to finite sequences of symbols of $\mathcal{L}$ (to *expressions of* $\mathcal{L}$) by setting

$$\#(s_0, \ldots, s_n) = \langle \mathrm{sn}(s_0), \ldots, \mathrm{sn}(s_n) \rangle\,,$$

where $\mathrm{sn}(s)$ is the symbol number of s. When we talk of the # of a symbol s, we mean $\#(s)$, i.e., $\langle \mathrm{sn}(s) \rangle$, which is $2^{\mathrm{sn}(s)+1}$. We assign numbers to sequences of expressions (for example, to deductions) by

$$\#(\psi_0, \ldots, \psi_n) = \langle \#\psi_0, \ldots, \#\psi_n \rangle\,.$$

Lemma 6.10. *The following are primitive recursive:*

(1) the set of all #'s of variables;

(2) the set of all #'s of terms;

(3) the set of all #'s of atomic formulas;

(4) the set of all #'s of formulas.

Proof. (1) For $a \in \mathbb{N}$, a is the # of a variable iff and only if

$$a \in \mathrm{Seq} \wedge \mathrm{lh}(a) = 1 \wedge 2 \text{ divides } (a)_0\,.$$

(2) Let f be the characteristic function of the set of all #'s of terms. We will show that $\bar{f}$ is primitive recursive, from which it follows that f is primitive recursive. Note first that $\bar{f}(0) = 1$. For any number a, a is the # of a term if and only if either a is the # of a variable or constant or

$$\begin{aligned}&(\exists b)(\exists c)(b < a \ \wedge \ c < {p_a}^{a \cdot \mathrm{lh}(a)} \ \wedge \ c \in \mathrm{Seq} \ \wedge \\ &\qquad b \text{ is the \# of a } \mathrm{lh}(c)\text{-place function symbol } \wedge \\ &\qquad (\forall i < \mathrm{lh}(c))((c)_i < a \ \wedge \ (c)_i \text{ is the \# of a term}) \ \wedge \\ &\qquad a = b * (\ast_{i<\mathrm{lh}(c)}(c)_i))\,.\end{aligned}$$

Because of the condition $(c)_i < a$, we can replace "$(c)_i$ is a term" by "$(\bar{f}(a))_{(c)_i} = 1$." Hence we can write $f(a)$ and so $\bar{f}(a+1)$ as a primitive recursive function of a and $\bar{f}(a)$. By (III), $\bar{f}$ is primitive recursive.

(3) is easy using (2).

The proof of (4) is similar in structure to that of (2). □

Lemma 6.11. *The set of all #'s of tautologies is primitive recursive.*

Proof. If ψ is a proper subformula of a formula φ, then $\#\psi < \#\varphi$. Using this fact, we can see that, for any $a \in \mathbb{N}$, a is the # of a tautology if and only if a is the # of a formula and, for all $e < {p_a}^{2(a+1)}$, if

$$\begin{aligned}&e \in \mathrm{Seq} \ \wedge \ \mathrm{lh}(e) = a + 1 \ \wedge \\ &\quad (\forall i \le a)\, (e)_i \le 1 \ \wedge \\ &\quad (\forall i \le a)(\forall j < i)(i = \#(\neg) * j \ \rightarrow \ (e)_i = 1 \dot{-} (e)_j) \ \wedge \\ &\quad (\forall i \le a)(\forall j < i)(\forall k < i)(i = \#(() * j * \#(\rightarrow) * k * \#()) \\ &\qquad\qquad\qquad\qquad \rightarrow \ (e)_i = \mathrm{sg}((1 \dot{-} (e)_j) + (e)_k))\,,\end{aligned}$$

then $(e)_a = 1$. □

Lemma 6.12. *(1) There is a primitive recursive function* Sb *such that, if φ is a formula or a term, x is a variable, and t is a term, then*

$$\mathrm{Sb}(\#\varphi, \#(x), \#t) = \#\varphi(t)$$

where $\varphi(t)$ is the result of substituting t for the free occurrences of x in φ.

(2) There is a primitive recursive relation Fr *such that, if φ is a formula and x is a variable, then*

$$\mathrm{Fr}(\#\varphi, \#(x)) \leftrightarrow \ x \textit{ occurs free in } \varphi\,.$$

(3) The set of all #'s of sentences is primitive recursive.

(4) There is a primitive recursive relation Sbl *such that, if* φ *is a formula and* x, t, *and* $\varphi(t)$ *are as in (1), then*

$$\mathrm{Sbl}(\#\varphi, \#(x), \#t) \leftrightarrow \text{ no occurrence of a variable in } t \text{ becomes bound in } \varphi(t).$$

Exercise 6.4. Prove Lemma 6.12

Hint. (1) Let $\mathrm{Sb}'(b, c, a) = \mathrm{Sb}(a, b, c)$. Define $\overline{\mathrm{Sb}'}$ by primitive recursion. (See the proof of part (2) of Lemma 6.10 for an illustration of the method.)

(2) What happens if you substitute **0** for a variable in a formula or term in which the variable does not occur free?

(4) Use part (2), and use primitive recursion as in part (1).

Lemma 6.13. *(a) The set of all #'s of logical axioms is primitive recursive.*

(b) The set of all $(\#\varphi, \#\psi, \#\chi)$ *such that* χ *follows from* φ *and* ψ *by Modus Ponens is primitive recursive.*

(c) The set of all $(\#\varphi, \#\psi)$ *such that* ψ *follows from* φ *by the Quantifier Rule is primitive recursive.*

Proof. (a) We have already dealt with tautologies in Lemma 6.11. The identity axioms are easily handled using parts (2) and (3) of Lemma 6.10 and the function Sb. Quantifier Axioms are handled using Sbl and Sb.

(b) and (c) are proved in a straightforward manner, with Fr used for the latter. □

Exercise 6.5. Prove part (b) of Lemma 6.13.

Lemma 6.14. *Suppose that* $\mathcal{L}$ *extends* $\mathcal{L}^{\mathrm{A}}$ *. The set of #'s of axioms of* PA *is primitive recursive.*

Proof. There are finitely many axioms plus the induction schema. Instances of the latter are easily characterized using Sb. □

A theory T in $\mathcal{L}$ is *recursively axiomatizable* if there is a set Σ of sentences such that

(i) $\{\#\sigma \mid \sigma \in \Sigma\}$ is recursive;

(ii) $\{\tau \mid T \models \tau\} = \{\tau \mid \Sigma \models \tau\}$.

The notion of a *primitively recursively axiomatizable* theory is similarly defined, with "primitive recursive" replacing "recursive" in clause (i).

Remark. In fact, the class of recursively axiomatizable theories turns out to be the same as the class of primitively recursively axiomatizable theories.

Lemma 6.15. *Suppose that T is a primitively recursively axiomatizable theory in $\mathcal{L}$. Let Σ witness this fact. Then there is a primitive recursive relation* Pr *such that, for all a and $b \in \mathbb{N}$,* $\mathrm{Pr}(a,b)$ *holds if and only if a is the # of a sentence τ and b is the # of a deduction of τ from Σ.*

Proof. The lemma follows easily from Lemma 6.13. □

Theorem 6.16. *The functions representable in* Q *are exactly the recursive functions.*

Proof. By Theorem 5.23, we need only show that every function representable in Q is recursive. Suppose $\varphi(v_1, \ldots, v_{n+1})$ represents $f : \mathbb{N}^n \to \mathbb{N}$ in Q. Let Pr be given by Lemma 6.15 for $T = \mathrm{Q}$ and for Σ our set of axioms for Q. Note that the function

$$(a_1, \ldots, a_{n+1}) \mapsto \#\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_{n+1}}\mathbf{0})$$

is primitive recursive, since the # of $\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_{n+1}}\mathbf{0})$ is

$$\mathrm{Sb}(\ldots(\mathrm{Sb}(\#\varphi, \#(v_1), \#\mathbf{S}^{a_1}\mathbf{0}), \ldots), \#(v_{n+1}), \#\mathbf{S}^{a_{n+1}}\mathbf{0}),$$

and since the function $a \mapsto \#\mathbf{S}^a\mathbf{0}$ is easily seen to be primitive recursive. Define a recursive function $g : \mathbb{N}^n \to \mathbb{N}$ by

$$g(a_1, \ldots, a_n) = \mu b\, \mathrm{Pr}(\#\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}, \mathbf{S}^{(b)_0}\mathbf{0}), (b)_1).$$

For all $(a_1, \ldots, a_n)$,

$$f(a_1, \ldots, a_n) = (g(a_1, \ldots, a_n))_0. \qquad \square$$

We now know that the recursive functions have all the closure properties of those representable in Q. (We could have directly proved those closure properties that we directly proved for the primitive recursive functions.) Thus we get the following lemma.

Lemma 6.17. *Lemma 6.15 continues to hold when the words "primitively" and "primitive" are deleted from its statement.*

Remark. By Lemma 6.17 and the proof of Lemma 6.16, any function representable in any recursively axiomatizable theory is recursive.

Lemma 6.18 (Fixed Point Lemma). *Let $\varphi(v_1)$ be a formula of $\mathcal{L}^{\mathrm{A}}$. There is a sentence σ such that*

$$\mathrm{Q} \models (\sigma \leftrightarrow \varphi(\mathbf{S}^{\#\sigma}\mathbf{0}))\,.$$

Proof. Let $\psi(v_1, v_2, v_3)$ represent in Q the primitive recursive function

$$(a, n) \mapsto \mathrm{Sb}(a, \#v_1, \#\mathbf{S}^n\mathbf{0})\,.$$

Note that, for any formula $\chi(v_1)$ and any $n \in \mathbb{N}$, this function sends $(\#\chi, n)$ to $\#\chi(\mathbf{S}^n\mathbf{0})$.

Let $\chi(v_1)$ be the following formula:

$$\forall v_3(\psi(v_1, v_1, v_3) \rightarrow \varphi(v_3))\,.$$

Let $q = \#\chi(v_1)$.

Now let σ be the sentence

$$\forall v_3(\psi(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \rightarrow \varphi(v_3))\,.$$

Note that σ is the result of replacing v_1 by $\mathbf{S}^q\mathbf{0}$ in the formula $\chi(v_1)$. In other words, $\#\sigma$ is the value of the function represented by ψ on the argument (q, q). Hence

$$\mathrm{Q} \models \forall v_3(\psi(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \;\leftrightarrow\; v_3 = \mathbf{S}^{\#\sigma}\mathbf{0})\,.$$

In particular,

$$\mathrm{Q} \models \psi(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^{\#\sigma}\mathbf{0})\,.$$

Thus

$$\mathrm{Q} \models (\sigma \rightarrow \varphi(\mathbf{S}^{\#\sigma}\mathbf{0})\,.$$

But also

$$\mathrm{Q} \models \forall v_3(\psi(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \;\rightarrow\; v_3 = \mathbf{S}^{\#\sigma}\mathbf{0})\,.$$

Therefore

$$\mathrm{Q} \models (\varphi(\mathbf{S}^{\#\sigma}\mathbf{0}) \rightarrow \sigma)\,. \qquad \square$$

It is worth recording the following fact: Suppose $\psi(v_1, \ldots, v_n)$ represents in Q a relation R. Since Q is true in $\mathfrak{N}$, we have that

$$(\forall a_1 \in \mathbb{N}) \cdots (\forall a_n \in \mathbb{N})(R(a_1, \ldots, a_n) \leftrightarrow \mathfrak{N} \text{ satisfies } \psi[a_1, \ldots, a_n]) .$$

Completeness of theories. A theory T in a Language $\mathcal{L}$ is *complete* if, for each sentence σ of $\mathcal{L}$, $T \models \sigma$ or $T \models \neg\sigma$.

Theorem 6.19. *Let T be a recursively axiomatizable theory in $\mathcal{L}^{\mathrm{A}}$ such that T is true in $\mathfrak{N}$. Then T is not complete.*

Proof. Let Pr be given by Lemma 6.17. Let ψ witness that Pr is representable in Q. Let $\varphi(v_1)$ be the formula

$$\forall v_2 \neg\psi(v_1, v_2) .$$

Let σ be given be the Fixed Point Lemma.

One can think of σ as expressing its own unprovability in T. Indeed, by the observation preceding the theorem,

$$T \not\models \sigma \leftrightarrow \sigma \text{ is true in } \mathfrak{N} .$$

Thus if $T \models \sigma$ then σ is false in $\mathfrak{N}$, contradicting the hypothesis that T is true in $\mathfrak{N}$. If $T \models \neg\sigma$ the fact that T is true in $\mathfrak{N}$ implies that σ is false in $\mathfrak{N}$, and this implies that the contradiction that $T \models \sigma$. □

Theorem 6.20. *Let T be any theory in $\mathcal{L}^{\mathrm{A}}$ such that $T \cup \mathrm{Q}$ is consistent. Then $\{\#\tau \mid T \models \tau\}$ is not recursive.*

Proof. Suppose for a contradiction that $\{\#\tau \mid T \models \tau\}$ is recursive. Let

$$T' = \{\tau \mid T \cup \mathrm{Q} \models \tau\} .$$

Let ρ be the conjunction of the finitely many axioms of Q. Then

$$\tau \in T' \;\leftrightarrow\; (\rho \rightarrow \tau) \in T ,$$

so $\{\#\tau \mid \tau \in T'\}$ is recursive.

By Theorem 5.23, let $\psi(v_1)$ represent $\{\#\tau \mid \tau \in T'\}$ in Q. Let σ be given by the Fixed Point Lemma with $\neg\psi$ as φ.

Suppose first that $\sigma \notin T'$. Then

$$\mathrm{Q} \models \neg\psi(\mathbf{S}^{\#\sigma}\mathbf{0}) .$$

But this implies that

$$\mathrm{Q} \models \sigma ,$$

which in turn implies that $\sigma \in T'$.

Suppose then that $\sigma \in T'$. We successively get that $\mathrm{Q} \models \psi(\mathbf{S}^{\#\sigma}\mathbf{0})$, that $\mathrm{Q} \models \neg\sigma$, and that $\neg\sigma \in T'$. □

Corollary 6.21 (Church's Theorem). *The set of all #'s of valid sentences in $\mathcal{L}^{\mathrm{A}}$ is not recursive.*

Corollary 6.22. *If T be a recursively axiomatizable theory in $\mathcal{L}^{\mathrm{A}}$ such that $T \cup \mathrm{Q}$ is consistent, then T is not complete.*

Proof. It suffices to prove that if Σ is a set of sentences such that $\{\#\sigma \mid \sigma \in \Sigma\}$ is recursive and the theory $T = \{\tau \mid \Sigma \models \tau\}$ is complete, then $\{\#\tau \mid \tau \in T\}$ is recursive. For this, fix Σ and let Pr be given by Lemma 6.17. Assume that T is is complete. Define $g : \mathbb{N} \to \mathbb{N}$ by setting $g(a) = 0$ if a is not the # of a sentence and otherwise setting

$$g(a) = \mu b\,(\mathrm{Pr}(a, b) \,\vee\, \mathrm{Pr}(\#(\neg\ ) * a, b))\,.$$

Since T is complete, g is a recursive function. Moreover, for any $a \in \mathbb{N}$,

$$a \in \{\#\tau \mid \tau \in T\} \,\leftrightarrow\, (g(a) \neq 0 \,\wedge\, \mathrm{Pr}(a, g(a)))\,. \qquad \square$$

A theory T in $\mathcal{L}$ is *recursively decidable* if $\{\#\tau \mid T \models \tau\}$ is recursive. Otherwise T is *recursively undecidable*. Thus Church's Theorem shows that the set of valid sentences of $\mathcal{L}^{\mathrm{A}}$ is not recursively decidable. (Church's Theorem is actually more general, holding for, say, any language with a two-place relation symbol.) According to *Church's Thesis* (also called the *Church-Turing Thesis*), the recursive functions are exactly the effectively computable functions. Granted Church's Thesis, decidability and recursive decidability are the same.

Theorem 6.23. PA *is incomplete and recursively undecidable. Moreover all recursively axiomatizable extensions of* PA *are incomplete, and all consistent extensions of* PA *are recursively undecidable.*

Proof. This follows from Theorem 6.19 or Corollary 6.22, and Theorem 6.20. □

Theorem 6.19, Theorem 6.20, Corollary 6.22, and Theorem 6.23 are all versions of Gödel's First Incompleteness Theorem. We end this section with a brief sketch of Gödel's Second Incompleteness Theorem.

Let Pr be given by Lemma 6.17 for some recursively axiomatizable T in $\mathcal{L}^{\mathrm{A}}$ such that $\mathrm{Q} \subseteq T$. Let ψ witness that Pr is representable in Q. Let σ be given by the Fixed Point Lemma, with $(\forall v_2)\neg\psi(v_1, v_2)$ as $\varphi(v_1)$. Thus $T \not\models \sigma$ if and only if σ is true in $\mathfrak{N}$.

Suppose that σ is false in $\mathfrak{N}$, i.e., suppose that $T \models \sigma$. Then there is a $b \in \mathbb{N}$ such that $\mathrm{Pr}(\#\sigma, b)$. For any such b,

$$\mathrm{Q} \models \psi(\mathbf{S}^{\#\sigma}\mathbf{0}, \mathbf{S}^{b}\mathbf{0})\,.$$

Hence

$$\mathrm{Q} \models (\exists v_2)\psi(\mathbf{S}^{\#\sigma}\mathbf{0}, v_2)\,.$$

In other words,

$$\mathrm{Q} \models \neg\varphi(\mathbf{S}^{\#\sigma}\mathbf{0})\,.$$

But then $\mathrm{Q} \models \neg\sigma$, and so $T \models \neg\sigma$. Therefore T is inconsistent.

The argument of the last paragraph shows that if T is consistent then σ is true in $\mathfrak{N}$. The converse of this fact also holds: If σ is true, then $T \not\models \sigma$, and so T is consistent. Thus σ is true in $\mathfrak{N}$ if and only if T is consistent.

Using the formula ψ and formulas representing the set of all #'s of sentences and the function $a \mapsto \#(\neg) * a$, we can construct a sentence $\ulcorner \mathrm{Con}\ T \urcorner$ of $\mathcal{L}^{\mathrm{PA}}$ that we may think of as expressing the consistency of T. Our argument then establishes the truth of

$$\sigma \leftrightarrow \ulcorner \mathrm{Con}\ T \urcorner\,.$$

Now comes the sketchy part of our discussion. If we have chosen natural representing formulas, then we can show that

$$\mathrm{PA} \models \sigma \leftrightarrow \ulcorner \mathrm{Con}\ T \urcorner\,.$$

This is essentially because our basic tool in our (presumably set theoretic) proof of (the set theoretic version of) this sentence was induction.

Now suppose that T is PA. Since PA is consistent, $\mathrm{PA} \not\models \sigma$. But then

$$\mathrm{PA} \not\models \ulcorner \mathrm{Con\ PA} \urcorner\,.$$

In other words, the consistency of PA implies that the number theoretic version of the consistency of PA is not provable in PA.

The argument establishes that any consisent, recursively axiomatizable extension of PA cannot prove the number-theoretic sentence expressing its own consistency. This result can easily be extended to theories in which PA is interpretable. For example, one cannot prove in ZFC, if ZFC is consistent, the set-theoretic formulation of the consistency of ZFC.

Exercise 6.6. Suppose we dropped the restriction that the variable x is does not occur free in φ from the Quantifier Rule. Would the modified deductive system be sound? Would it be complete? Prove your answers.

Exercise 6.7. Let $\mathcal{L} = \{\sim\}$. Let $\mathfrak{A}$ be a model in which $\sim_{\mathfrak{A}}$ is an equivalence relation which has one equivalence class of size n for each natural number $n > 0$. Prove that there is a model $\mathfrak{B}$ such that $\mathrm{Th}(\mathfrak{A}) = \mathrm{Th}(\mathfrak{B})$ and such that $\mathfrak{B}$ has an infinite equivalence class. (Recall that $\mathrm{Th}(\mathfrak{A})$ is the set of all sentences true in $\mathfrak{A}$.)

Exercise 6.8. Let $\mathcal{L}$ be a language. Let Σ be a set of sentences of $\mathcal{L}$ and let τ be a sentence of $\mathcal{L}$. P be a one-place relation symbol of $\mathcal{L}$ that does not occur in Σ or in τ. Assume that $\Sigma \vdash \tau$. Prove that there is a deduction in $\mathcal{L}$ of τ from Σ in which P does not occur.