Philosophy 135 Winter 2007 Tony Martin

Introduction to Metalogic

1 The semantics of sentential logic.

The language $\mathcal{L}$ of sentential logic.

Symbols of $\mathcal{L}$:

(i)	sentence letters	$p_0, p_1, p_2, \ldots$
(ii)	connectives	$\neg, \vee$
(iii)	parentheses	(,)

Remarks:

(a) We shall pay little or no attention to the use-mention distinction. For instance, we are more likely to write "p_1 is a sentence letter" than "'p_1' is a sentence letter."

(b) There are several standard variants of our list of connectives. Trivial variants can be gotten by using literally different symbols to play the roles ours play. For example, it is common to use $\sim$ in place of our $\neg$. Other variants can be gotten by using additional symbols that play different roles from those ours play, e.g., connectives $\wedge$, $\rightarrow$, and $\leftrightarrow$. We do not do this, in order to keep definitions and proofs as short and simple as possible. We will, however, introduce the symbols mentioned above as *abbreviations.* Instead of adding connectives to our list, one could replace our connectives with others. For example, one could drop $\vee$ and replace it by $\wedge$ We shall occasionally make remarks on how such changes would affect our definitions of semantic and deductive concepts.

Formulas of $\mathcal{L}$:

(i) Each sentence letter is a formula.

(ii) If A is a formula, then so is $\neg A$.

(iii) If A and B are formulas, then so is $(A \vee B)$.

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

Let us officially regard formulas as sequences of symbols. Thus the formula $(p_1 \vee \neg p_2)$ is officially a sequence of length 6. This official stance will make little practical difference.

We often want to prove that all formulas have some property P. A method for proving this is *formula induction.* To prove by formula induction that every formula has property P, we must prove (i), (ii), and (iii) below.

(i) Each sentence letter has property P.

(ii) If A is a formula that has property P, then $\neg A$ has P.

(iii) If A and B are formulas that have property P, then $(A \vee B)$ has P.

If we prove (i)-(iii) for P, then clause (iv) in the definition of formulas guarantees that all formulas have property P.

The proof of the following lemma is an example of proof by formula induction.

Lemma 1.1. *Every formula contains the same number of (occurrences of) left parentheses as (occurrences of) right parentheses.*

Proof. Let P be the property of being a formula with the same number of left as right parentheses.

(i) Sentence letters have no parentheses, so clearly they have property P.

(ii) Assume that A is a formula and that A has P. Since $\neg A$ has the same occurrences of left and right parentheses as does A, $\neg A$ has P.

(iii) Assume that A and B are formulas having property P. The number of left parentheses in $(A \vee B)$ is $m + n + 1$, where m is the number of left parentheses in A and n is the number of left parentheses in B, and the number of right parentheses in $(A \vee B)$ is $m' + n' + 1$, where m' is the number of right parentheses in A and n' is the number of right parentheses in B. By assumption, $m = m'$ and $n = n'$; so $m + n + 1 = m' + n' + 1$. Thus $(A \vee B)$ has P.

The lemma follows by formula induction. $\square$

Lemma 1.2. *For every formula A, exactly one of the following holds.*

(1) A is a sentence letter.

(2) There is a formula B such that A is $\neg B$.

(3) There are formulas B and C such that A is $(B \vee C)$.

Proof. Evidently at most one of (1)–(3) can hold for any formula, so we need only show that for each formula at least one of (1)–(3) holds. Since all the formulas given by instances of clauses (i)–(iii) in the definition of formula are of these forms, the desired conclusion follows by clause (iv).

Lemma 1.3. *For every formula A,*

(a) every initial segment of A has the at least as many left as right parentheses;

(b) if A is a disjunction (i.e., is $(B \vee C)$ for some formulas B and C), then every proper initial segment of A (i.e., every initial segment of A that is not the whole of A) that has length greater than 0 *has more left than right parentheses.*

Proof. Let P be the property of being a formula for which (a) and (b) hold. We use formula induction to prove that all formulas have P. In each of steps (i) and (ii), the proof that (a) holds is similar to the corresponding step of the proof of Lemma 1.1. For steps (i) and (ii), (b) holds vacuously. We need then only prove that (b) holds for $(A \vee B)$ on the assumption that A and B have property P. Let C be a proper initial segment of $(A \vee B)$ of length greater than 0. C contains the initial (and does not contain the final (. The desired conclusion follows from the assumption that (a) holds for A and B. □

Lemma 1.4. *No proper initial segment of a formula is a formula.*

Proof. We use formula induction, with P the property of being a formula no proper initial segment of which is a formula.

Note that (i) is trivial. Note also that (iii) follows from Lemmas 1.3 and 1.1. This is because part (b) of Lemma 1.3 says that non-zero length proper initial segments of disjunctions have more left than right parentheses, while Lemma 1.1 says that formulas have the same number of left as right parentheses.

For (ii), assume that A has P. Let D be a proper initial segment of $\neg A$. Since the empty sequence is not a formula, we may assume that D has length > 0. Thus D is $\neg A'$, where A' is a proper initial segment of A. Since A has P, A' is not a formula. It follows from this fact and Lemma 1.2 that $\neg A'$ is not a formula. □

Theorem 1.5 (Unique Readability). *Let A be a formula. Then exactly one of the following holds.*

(1) A is sentence letter.

(2) There is a unique formula B such that A is $\neg B$.

(3) There are unique formulas B and C such that A is $(B \vee C)$.

Proof. If A does not begin with a left parenthesis, then Lemma 1.2 implies that exactly one of (1) or (2) holds.

Assume that A begins with a left parenthesis. Then there must be formulas B and C such that A is $(B \vee C)$. Assume that there are formulas B' and C' such that B' is different from B and A is $(B' \vee C')$. Then one of B and B' must be a proper initial segment of the other, contradicting Lemma 1.4. □

Exercise 1.1. Prove by formula induction that, for every formula A, the number of occurrences of sentence letters in A is one more than the number of occurrences of $\vee$ in A.

Truth and logical implication.

We now know that our language has an unambiguous grammar. Our next task is to introduce for it semantic notions such as meaning and truth. The natural way to proceed is from the bottom up: first to give meanings to the sentence letters; then to give meanings to the connectives and to use this to give meanings—and truth conditions—to the formulas of $\mathcal{L}$.

Let us first consider the sentence letters. As the name suggests, they are to be treated as whole (declarative) sentences. To give them a meaning, we should specify what statement or proposition each of them expresses. (Sentential logic is sometimes called *propositional logic* and sentence letters are sometimes called *proposition letters*.) One way to do this would be to assign to each sentence letter a declarative sentence of English whose *translation* it would be. The sentence letter would then have the same meaning, express the same proposition, as the English sentence.

If we did what was just suggested, then each sentence letter would be given a meaning once and for all. Once we specified the meanings of the connectives, then $\mathcal{L}$ would be a language in the usual sense, albeit an artificial and a very simplified one. But we do not want to use $\mathcal{L}$ in this way, to express particular propositions. Instead we want to use it to study logical relations between propositions, to study relations between propositions that depend only on the logical forms of the propositions. Therefore we shall not specify a fixed way of assigning a proposition to each sentence letter, but we shall try to consider all ways in which this might be done, all ways in which the language could be turned into a language in the usual sense.

We want to define the general notion of what we might call an *interpretation* of $\mathcal{L}$ or a *model* for $\mathcal{L}$, but what we shall actually call a *valuation* for $\mathcal{L}$. We could define a valuation to be an assignment of a declarative English sentence to each sentence letter. This seems, however, too restrictive a notion, since there are surely many propositions that are not expressed by any English sentence. We could instead define a valuation as an assignment of a proposition to each sentence letter. But we shall have no reason to be concerned with the *content* of the propositions assigned to the sentence letters. We shall only need to deal with their *truth-values*, with whether or not they are true or false. Because we shall be doing *truth-functional* logic, the truth conditions for complex formulas will depend only on the truth-values of the sentence letters that occur in them, and not on what propositions the sentence letters express.

We define then a *valuation* v for $\mathcal{L}$ to be a *function* that assigns to each sentence letter of $\mathcal{L}$ a *truth-value* $\mathbf{T}$ or $\mathbf{F}$.

Let v be a valuation for $\mathcal{L}$. The valuation v directly gives us a truth-value to each sentence letter. We next describe how it indirectly gives a truth-value to each formula of $\mathcal{L}$. To do this we define a function v^* that assigns a truth-value to each formula of $\mathcal{L}$, so that

(a) if A is a sentence letter, then $v^*(A) = v(A)$;

(b) $v^*(\neg A) = \begin{cases} \mathbf{F} & \text{if } v^*(A) = \mathbf{T}\,; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F}\,; \end{cases}$

(c) $v^*((A \vee B)) = \begin{cases} \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{F}\,; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{F}\,. \end{cases}$

We define a formula A to be *true* under the valuation v if $v^*(A) = \mathbf{T}$ and to be *false* under v if $v^*(A) = \mathbf{F}$.

Have we actually defined the function v^*? We have, for each of the three kinds of formulas, told by an equation what v^* assigns to formulas of that kind. But "v^*" appears on the right side as well as on the left side of these equations, so this is not an ordinary definition. It is what is called a *recursive* or *inductive* definition.

An example will make it intuitively clear that clauses (a)–(c) determine what truth-value v^* assigns to any given formula. Consider

$$(\neg p_3 \vee \neg(p_1 \vee p_3)).$$

Assume that $v(p_1) = \mathbf{T}$ and $v(p_3) = \mathbf{F}$. Then

$$\begin{aligned}
v^*(p_1) &= \mathbf{T} && \text{(by (a));}\\
v^*(p_3) &= \mathbf{F} && \text{(by (a));}\\
v^*(\neg p_3) &= \mathbf{T} && \text{(by (b));}\\
v^*((p_1 \vee p_3)) &= \mathbf{T} && \text{(by (c));}\\
v^*(\neg(p_1 \vee p_3)) &= \mathbf{F} && \text{(by (b));}\\
v^*((\neg p_3 \vee \neg(p_1 \vee p_3))) &= \mathbf{T} && \text{(by (c)).}
\end{aligned}$$

Thus $(\neg p_3 \vee \neg(p_1 \vee p_3))$ is true under v.

The definition of v^* is an example of *definition by recursion on formulas.* This is a method for defining a function h whose domain is the set of all formulas. To define h by this method, one must

(a) define $h(A)$ from A for sentence letters A;

(b) define $h(\neg A)$ from A and $h(A)$ for formulas A;

(c) define $h((A \vee B))$ from A, B, $h(A)$, and $h(B)$ for formulas A and B.

Here we are being a little imprecise in order to be comprehensible. Remaining at the same level of imprecision, let us sketch how to use formula induction to prove that doing (a)–(c) determines a unique function h whose domain is the set of all formulas. Suppose (a)–(c) have been done. Let P be the property of being a formula A for which a unique value $h(A)$ is determined by the definitions of (a)–(c). For (i) and (ii), use the definitions of (a) and (b) and the trivial parts of Unique Readability. For (iii), assume that A and B are formulas that have P. The definition of (c) determines a value of $h((A \vee B))$ from the values of $h(A)$ and $h(B)$ given by the fact that A and B have P. The uniqueness of this value follows from the uniqueness of $h(A)$ and $h(B)$ together with Unique Readability.

It will be convenient to make to introduce some abbreviations:

$$\begin{aligned}
&(A \wedge B) && \text{for} && \neg(\neg A \vee \neg B);\\
&(A \to B) && \text{for} && (\neg A \vee B);\\
&(A \leftrightarrow B) && \text{for} && ((A \to B) \wedge (B \to A)).
\end{aligned}$$

Bear in mind that $\wedge$, $\to$, and $\leftrightarrow$ are not actually symbols of $\mathcal{L}$. 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.

Let us also consider $\supset$ as an "abbreviation" for $\to$ and $\sim$ as an "abbreviation" for $\neg$ (since some students may be more used to these symbols than to the official ones).

It is not hard to see that the defined symbols $\wedge$, $\rightarrow$, and $\leftrightarrow$ obey the following rules:

$$\text{(d)}\;\; v^*((A \wedge B)) \;=\; \begin{cases} \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{F}\,; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{F}\,; \end{cases}$$

$$\text{(e)}\;\; v^*((A \rightarrow B)) \;=\; \begin{cases} \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{F}\,; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{F}\,; \end{cases}$$

$$\text{(f)}\;\; v^*((A \leftrightarrow B)) \;=\; \begin{cases} \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{F}\,; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{T}\,; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{F}\,. \end{cases}$$

Exercise 1.2. Let v be the valuation for $\mathcal{L}$ defined as follows.

$$v(p_i) \;=\; \begin{cases} \mathbf{T} & \text{if } i \text{ is even;} \\ \mathbf{F} & \text{if } i \text{ is odd.} \end{cases}$$

Using the tables above, determine which of the following two formulas are true under v.

(1) $(p_1 \leftrightarrow (\neg p_1 \vee p_1))\,;$

(2) $((p_0 \rightarrow p_3) \rightarrow (\neg p_5 \rightarrow \neg p_4))\,.$

Exercise 1.3. Prove that the formula $(\neg\neg p_0 \leftrightarrow p_0)$ is true under v for every valuation v for $\mathcal{L}$.

Exercise 1.4. Use definition by recursion on formulas to define a function h such that, for every formula A, $h(A)$ is the first sentence letter occurring in A.

Let Γ be a set of formulas of $\mathcal{L}$ and let A be a formula of $\mathcal{L}$. Consider the *argument* $\Gamma \mathrel{\therefore} A$ with (set of) *premises* Γ and *conclusion* A. We say that this argument is *valid* if the formula A is true under every valuation v for $\mathcal{L}$ such that all the formulas in Γ are true under v. To express this more briefly, let us say that a set of formulas is *true* under a valuation v if all the formulas belonging to the set are true under v. Then $\Gamma \mathrel{\therefore} A$ is a valid argument if and only if A is true under every valuation under which Γ is true.

There is a different way to talk about valid arguments, and we shall usually talk in this second way. If Γ is a set of formulas and A is a formula, then say that Γ *logically implies* A if $\Gamma \therefore A$ is a valid argument. We write $\Gamma \models A$ to mean that Γ logically implies A.

A special case of valid arguments and logical implication occurs when Γ is the empty set $\emptyset$. We usually write $\models A$ instead of $\emptyset \models A$. When $\models A$ we say that A is *valid* or that A is a *tautology*. A formula is a tautology if and only if it is true under every valuation for $\mathcal{L}$.

A formula is *satisfiable* if it is true under some valuation. Similarly a set of formulas is *satisfiable* if it is true under some valuation, i.e., if there is a valuation under which all the formulas in the set are true.

Exercise 1.5. Which of the following are tautologies? Prove your answers.

(1) $((p_0 \to (p_1 \to p_2)) \to (p_1 \to p_2))\,.$

(2) $((p_0 \to p_1) \vee (p_1 \to p_2))\,.$

Exercise 1.6. Which of the following are statements are true? Prove your answers.

(1) $\{(p_0 \to \neg p_1),\ ((p_2 \vee p_0) \to (p_1 \vee p_2)),\ \neg p_2\} \ \models\ \neg p_0\,.$

(2) $\{((\neg p_3 \vee p_0) \vee p_1),\ (\neg p_1 \to \neg p_2),\ (p_0 \to (p_2 \wedge p_3))\} \ \models\ p_1\,.$

If A and B are formulas, then by $A \models B$ we mean that $\{A\} \models B$.

Exercise 1.7. Let Γ and Δ be sets of formulas and let A, B, and $A_1, \ldots, A_n$ be formulas. Prove each of the following.

(1) $\Gamma \cup \{A\} \models B$ if and only if $\Gamma \models (A \to B)\,.$

(2) $\{A_1, \ldots, A_n\} \models B$ if and only if $\models (A_1 \to \cdots \to A_n \to B)\,.$

(3) A is satisfiable if and only if $A \not\models (p_0 \wedge \neg p_0)\,.$

(4) If $\Gamma \models C$ for every C belonging to Δ and if $\Delta \models B$, then $\Gamma \models B$.

When we omit parentheses in a formula, as we did in (2), we make use of a convention that omitted parentheses group to the right. Thus $(A_1 \to \cdots \to A_n \to B)$ abbreviates $(A_1 \to (\cdots \to (A_n \to B) \cdots)$.

Mathematical induction: To prove that all natural numbers have some property P, one may use *mathematical induction*. To do this one must prove (i) and (ii) below.

(i) 0 has P.

(ii) If n is a natural number that has P, then $n+1$ has P.

One can define functions by *definition by recursion on natural numbers* as well as by recursion on formulas. Recursion on natural numbers is a method for defining a function h whose domain is the set $\mathbf{N}$ of all natural numbers. To define h by this method, one must

(a) define $h(0)$;

(b) define $h(n+1)$ from n and $h(n)$ for natural numbers n.

Example. The clauses

(i) $h(0) = 0$;

(ii) $h(n+1) \; = \; h(n)+1+1$;

give a definition by recursion of the doubling function (in terms of the successor function $+1$).

Exercise 1.8. The *factorial* function is the function h with domain $\mathbf{N}$ such that $h(0) = 1$ and, for every $n > 0$, $h(n)$ is the product of all the positive integers $\leq n$. Show how to define the factorial function by recursion on natural numbers.

We now embark on the proof of the Compactness Theorem, one of the main theorems about our semantics for $\mathcal{L}$. Say that a set Γ of formulas is *finitely satisfiable* if every finite subset of Γ is satisfiable. The Compactness Theorem will assert that every finitely satisfiable set of formulas is satisfiable.

Lemma 1.6. *Let Γ be a finitely satisfiable set of formulas and let A be a formula. Then either $\Gamma \cup \{A\}$ is finitely satisfiable or $\Gamma \cup \{\neg A\}$ is finitely satisfiable.*

Proof. Assume for a contradiction neither $\Gamma \cup \{A\}$ nor $\Gamma \cup \{\neg A\}$ is finitely satisfiable. It follows that there are finite subsets Δ and Δ' of Γ such that neither $\Delta \cup \{A\}$ nor $\Delta' \cup \{\neg A\}$ is satisfiable. Since Γ is finitely satisfiable, the finite subset $\Delta \cup \Delta'$ of Γ is satisfiable. Let v be a valuation under which $\Delta \cup \Delta'$ is true. If A is true under v, then $\Delta \cup \{A\}$ is true under v and so is satisfiable. Otherwise $\Delta' \cup \{\neg A\}$ is true under v and is satisfiable. In either case we have a contradiction. $\square$

Lemma 1.7. *Let Γ be a finitely satisfiable set of formulas. There is a set Γ^* of formulas such that*

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

(2) Γ^* *is finitely satifiable;*

(3) for every formula A*, either* A *belongs to* Γ^* *or* $\neg A$ *belongs to* Γ^**.*

Proof. We can list all the formulas in an infinite list as follows. Think of the symbols of $\mathcal{L}$ as forming an infinite "alphabet" with the alphabetical order

$$\neg,\ \vee,\ (,\ ),\ p_0,\ p_1,\ p_2,\ \ldots .$$

First list in alphabetical order all the (finitely many) formulas that have length 1 and contain no occurrences of sentence letters other than p_0. Next list in alphabetical order all the remaining formulas that have length ≤ 2 and contain no occurrences of sentence letters other than p_0 and p_1. Next list in alphabetical order all the remaining formulas that have length ≤ 3 and contain no occurrences of sentence letters other than p_0, p_1, and p_2. 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 A_n in following paragraph.)

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

$$A_0,\ A_1,\ A_2,\ A_3,\ \ldots .$$

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

Let $\Gamma_0 = \Gamma$.

Let

$$\Gamma_{n+1} \;=\; \begin{cases} \Gamma_n \cup \{A_n\} & \text{if } \Gamma_n \cup \{A_n\} \text{ is finitely satisfiable;} \\ \Gamma_n \cup \{\neg A_n\} & \text{otherwise.} \end{cases}$$

Let $\Gamma^* = \bigcup_n \Gamma_n$.

Because $\Gamma = \Gamma_0 \subseteq \Gamma^*$, Γ^* has property (1).

Γ_0 is finitely satisfiable. By Lemma 1.6, if Γ_n is finitely satisfiable then so is Γ_{n+1}. By mathematical induction, every Γ_n is finitely satisfiable. If Δ is a finite subset of Γ^*, then $\Delta \subseteq \Gamma_n$ for some n. Since Γ_n is finitely satisfiable, Δ is satisfiable. Thus Γ^* has property (2).

Because either A_n or $\neg A_n$ belongs to Γ_{n+1} for each n and because each $\Gamma_{n+1} \subseteq \Gamma^*$, Γ^* has property (3). □

It will be convenient to introduce the symbol "$\in$" as an abbreviation for "belongs to."

Lemma 1.8. *Let Γ^* be a set of formulas having properties (2) and (3) described in the statement of Lemma 1.7. Then Γ^* is satisfiable.*

Proof. Define a valuation v for $\mathcal{L}$ by setting

$$v(A) = \mathbf{T} \text{ if and only if } A \in \Gamma^*$$

for each sentence letter A. Let P be the property of being a formula A such that

$$v^*(A) = \mathbf{T} \text{ if and only if } A \in \Gamma^* .$$

We prove by formula induction that every formula has property P.

(i) For sentence letters, this is true by definition of v.

(ii) First we show that $\neg A \in \Gamma^*$ if and only if $A \notin \Gamma^*$ for any formula A. By (3) we have that $A \in \Gamma^*$ or $\neg A \in \Gamma^*$. Suppose that both A and $\neg A$ belong to Γ^*. Then $\{A, \neg A\}$ is a finite subset of Γ^*. By (2) we get the contradiction that $\{A, \neg A\}$ is satisfiable.

Now let A be a formula that has property P. Then

$$\begin{aligned} v^*(\neg A) = \mathbf{T} \quad & \text{if and only if} \quad v^*(A) = \mathbf{F} \\ & \text{if and only if} \quad A \notin \Gamma^* \\ & \text{if and only if} \quad \neg A \in \Gamma^* . \end{aligned}$$

(iii) We first show that $(A \vee B) \in \Gamma^*$ if and only if either $A \in \Gamma^*$ or $B \in \Gamma^*$, for any formulas A and B. Assume first that $(A \vee B) \in \Gamma^*$ but that $A \notin \Gamma^*$ and $B \notin \Gamma^*$. By (3), $\neg A \in \Gamma^*$ and $\neg B \in \Gamma^*$. Thus $\{(A \vee B), \neg A, \neg B\}$ is a finite subset of Γ^*. By (2) we get the contradiction that $\{(A \vee B), \neg A, \neg B\}$ is satisfiable. Next assume that $A \in \Gamma^*$ but $(A \vee B) \notin \Gamma^*$. By (3) $\neg(A \vee B) \in \Gamma^*$, and so $\{A, \neg(A \vee B)\}$ is a finite subset of Γ^*. By (2) we get the contradiction that $\{A, \neg(A \vee B)\}$ is satisfiable. A similar argument shows that if $B \in \Gamma^*$ then $(A \vee B) \in \Gamma^*$.

Now let A and B be formulas that have property P. Then

$$\begin{aligned} v^*((A \vee B)) = \mathbf{T} \quad & \text{if and only if} \quad v^*(A) = \mathbf{T} \text{ or } v^*(B) = \mathbf{T} \\ & \text{if and only if} \quad A \in \Gamma^* \text{ or } B \in \Gamma^* \\ & \text{if and only if} \quad (A \vee B) \in \Gamma^* . \end{aligned}$$

Since, in particular, $v^*(A) = \mathbf{T}$ for every member of A of Γ^*, we have shown that Γ^* is satisfiable. $\square$

Exercise 1.9. Suppose that we added $\wedge$ as an official symbol of $\mathcal{L}$, extending the definition of truth using the table for $\wedge$ on page 7. Then proof by formula induction would have an extra step: showing that $(A \wedge B)$ has property P if both A and B have P. Supply this $(A \wedge B)$ case for the proof by formula induction just given.

Theorem 1.9 (Compactness). *Let Γ be a finitely satisfiable set of formulas. Then Γ is satisfiable.*

Proof. By Lemma 1.7, let Γ^* have properties (1)–(3) of that lemma. By Lemma 1.8, Γ^* is satisfiable. Hence Γ is satisfiable. $\square$

Corollary 1.10 (Compactness, Second Form). *Let Γ be a set of formulas and let A be a formula such that $\Gamma \models A$. Then there is a finite subset Δ of Γ such that $\Delta \models A$.*

Exercise 1.10. Prove Corollary 1.10.

2 Deduction in Sentential Logic

Though we have not yet introduced any formal notion of deductions (i.e., of derivations or proofs), we can easily give a formal method for showing that formulas are tautologies: Construct the truth table of a given formula; i.e., compute the truth-value of the formulas for all possible assignments of truth-values to the sentence letters occurring in it. If all these truth values are **T**, then the formula is a tautology. This method extends to give a formal method for showing that $\Gamma \models A$, provided that Γ is finite. The method even extends to the case Γ is infinite, since the second form of Compactness guarantees that if $\Gamma \models A$ then $\Delta \models A$ for some finite $\Delta \subseteq \Gamma$.

Nevertheless we are now going to introduce a different system of formal deduction. This is because we want to gain experience with the metatheory of a more standard deductive system.

The system SL.

Axioms: From now on we shall often adopt the convention of omitting outmost parentheses in formulas. For any formulas A, B, and C, each of the following is an *axiom* of our deductive sytem.

(1) $A \to (A \vee B)$

(2) $B \to (A \vee B)$

(3) $(A \vee B) \to (\neg A \to B)$

(4) $(\neg A \to B) \to ((\neg A \to \neg B) \to A)$

(5) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

Remarks:

(a) Note that (1)–(5) are not axioms but *axiom schemas*. There are infinitely many instances of each of these schemas, since A, B, and C may be any formulas whatsoever.

(b) Note also that we have used abbreviations in presenting these axiom schemas. For example, the (except for outer parentheses) unabbreviated Axiom Schema (1) is $\neg A \vee (A \vee B)$.

Rule of Inference:

$$\text{Modus Ponens (MP)} \qquad \frac{A\,,\ (A \to B)}{B}$$

For any formulas A and B, we say that B *follows by modus ponens* from A and $(A \to B)$.

Deductions: A *deduction* in **SL** from a set Γ of formulas is a finite sequence **D** of formulas such that whenever a formula A occurs in the sequence **D** then at least one of the following holds.

(1) $A \in \Gamma$.

(2) A is an axiom.

(3) A follows by modus ponens from two formulas occurring earlier in the sequence **D**.

If A is the nth element of the sequence **D**, then we say that A *is on line* n *of* **D** or even that A *is line* n *of* **D**.

A *deduction in* **SL** *of* A *from* Γ is a deduction **D** in **SL** from Γ with A on the last line of **D**. We write $\Gamma \vdash_{\mathbf{SL}} A$ and say A is *deducible* in **SL** from Γ to mean that there is a deduction in **SL** of A from Γ. Sometimes we may express this by saying Γ *proves* A *in* **SL**. We write $\vdash_{\mathbf{SL}} A$ for $\emptyset \vdash_{\mathbf{SL}} A$. We shall mostly omit the subscript "**SL**" and the phrase "in **SL**" during our study of sentential logic, since **SL** will be the only system we consider until we get to predicate logic.

Example 1. Let A and B be any formulas. Here is a very short deduction of $A \to (B \to A)$ from $\emptyset$. This deduction shows that $\vdash A \to (B \to A)$.

1.	$A \to (B \to A)$	Ax. 2
	$[A \to (\neg B \vee A)]$	

In square brackets we have rewritten line 1 in a less abbreviated way, in order to show that it is an instance of Axiom Schema 2. The formula A is the B of the schema, and the formula $\neg B$ is the A of the schema.

Example 2. Below we give a deduction of $A \to A$ from $\emptyset$. This deduction shows that $\vdash A \to A$.

1.	$(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$	Ax. 5
2.	$A \to ((A \to A) \to A)$	Ax. 2
3.	$(A \to (A \to A)) \to (A \to A)$	1,2; MP
4.	$A \to (A \to A)$	Ax. 2
5.	$A \to A$	3,4; MP

Theorem 2.1 (Deduction Theorem). *Let Γ be a set of formulas and let A and B be formulas. If $\Gamma \cup \{A\} \vdash B$ then $\Gamma \vdash (A \to B)$.*

Proof. Assume that $\Gamma \cup \{A\} \vdash B$. Let $\mathbf{D}$ be a deduction of B from $\Gamma \cup \{A\}$. We prove that

$$\Gamma \vdash (A \to C)$$

for every line C of $\mathbf{D}$. Assume that this is false. Consider the first line C of $\mathbf{D}$ such that $\Gamma \nvdash (A \to C)$.

Assume that C either belongs to Γ or is an axiom. The following gives a deduction of $(A \to C)$ from Γ.

1.	C	
2.	$C \to (A \to C)$	Ax. 2
3.	$A \to C$	1,2; MP

Assume next that C is A. We have already shown that $\vdash (A \to A)$. Thus $\Gamma \vdash (A \to A)$.

Finally assume that C follows from formulas E and $(E \to C)$ by MP. These formulas are on earlier lines of $\mathbf{D}$ than C. Since C is the first "bad" line of $\mathbf{D}$, let $\mathbf{D}_1$ be a deduction of $(A \to E)$ from Γ and let $\mathbf{D}_2$ be a deduction of $(A \to (E \to C))$ from Γ. We get a deduction of $(A \to C)$ from Γ by beginning with $\mathbf{D}_1$, following with $\mathbf{D}_2$, and then finishing with the lines

$(A \to (E \to C)) \to ((A \to E) \to (A \to C))$	Ax. 5
$(A \to E) \to (A \to C)$	MP
$A \to C$	MP

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 $\Gamma \vdash (A \to B)$. $\square$

Remarks:

(a) The converse of the Deduction Theorem is also true. Given a deduction of $(A \to B)$ from Γ, one gets a deduction of B from $\Gamma \cup \{A\}$ by appending the lines A and B, the latter coming by MP.

(b) The proof of the Deduction Theorem would still go through if we added or dropped axioms, as long as we did not drop Axiom Schemas 2 and 5. It would not in general go through if we added rules of inference, and it would not go through if we dropped the rule of modus ponens.

Exercise 2.1. Show that the following hold for all formulas A and B.

(a) $\vdash (A \to (\neg A \to B))$;

(b) $\vdash (\neg\neg A \to A)$.

A set Γ of formulas is *inconsistent* (in **SL**) if there is a formula B such that $\Gamma \vdash B$ and $\Gamma \vdash \neg B$. Otherwise Γ is *consistent*.

Theorem 2.2. *Let Γ and Δ be sets of formulas and let A, B, and $A_1, \ldots, A_n$ be formulas.*

(1) $\Gamma \cup \{A\} \vdash B$ if and only if $\Gamma \vdash (A \to B)$.

(2) $\Gamma \cup \{A_1, \ldots, A_n\} \vdash B$ if and only if $\Gamma \vdash (A_1 \to \ldots \to A_n \to B)$.

(3) Γ is consistent if and only if there is some formula C such that $\Gamma \nvdash C$.

(4) If $\Gamma \vdash C$ for all $C \in \Delta$ and if $\Delta \vdash B$, then $\Gamma \vdash B$.

Proof. We begin with (4). Let **D** be a deduction of B from Δ. We can turn **D** into a deduction of B from Γ as follows: whenever a formula $C \in \Delta$ is on a line of **D**, replace that line with a deduction of C from Γ.

(1) is just the combination of the Deduction Theorem and its converse.

For (2), forget the particular Γ, $A_1, \ldots, A_n$, and B for the moment and let P be the property of being a positive integer n such that (2) holds for every choice of Γ, $A_1, \ldots, A_n$, and B. By a variant of mathematical induction (beginning with 1 instead of with 0) we show that every positive integer has P. The integer 1 has P by (1). Assume that n is a positive integer that has P. Let Γ, $A_1, \ldots, A_{n+1}$, and B be given. By (1) we have that

$$\Gamma \cup \{A_1, \ldots, A_{n+1}\} \vdash B \text{ if and only if } \Gamma \cup \{A_1, \ldots, A_n\} \vdash (A_{n+1} \to B).$$

Since n has P, this holds if and only if $\Gamma \vdash (A_1 \to \ldots \to A_{n+1} \to B)$.

For the "if" part of (3), assume that Γ is inconsistent. Let B be such that $\Gamma \vdash B$ and $\Gamma \vdash \neg B$. Let C be any formula. Using Axiom Schema 2 and MP, we get that $\Gamma \vdash (\neg C \to B)$ and $\Gamma \vdash (\neg C \to \neg B)$. The formula

$$(\neg C \to B) \;\to\; ((\neg C \to \neg B) \;\to\; C)$$

is an instance of Axiom Schema 4. Two applications of MP show that $\Gamma \vdash C$.

The "only if" part of (3) is obvious. $\square$

Lemma 2.3. *For any formulas A and B,*

(a) $\{(\neg A \to B)\} \ \vdash \ (\neg B \to A)$ *;*

(b) $\{(A \to B)\} \ \vdash \ (\neg B \to \neg A)$ *.*

Proof. (a) By the Deduction Theorem, it is enough to show that

$$\{(\neg A \to B)\,,\ \neg B\} \ \vdash \ A\,.$$

Let $\Gamma = \{(\neg A \to B)\,,\ \neg B\}$. Axiom Schema 2 and MP give that $\Gamma \vdash (\neg A \to \neg B)$. The formula

$$(\neg A \to B) \ \to \ ((\neg A \to \neg B) \ \to \ A)$$

is an instance of Axiom Schema 4. Two applications of MP show that $\Gamma \vdash A$.

(b) Since $\vdash (\neg\neg A \to A)$ by part (b) of Exercise 2.1, we can use the Deduction theorem and easily get that

$$\{(A \to B)\} \vdash (\neg\neg A \to B)\,.$$

But $\{(\neg\neg A \to B)\} \vdash (\neg B \to \neg A)$ by part (a). □

Exercise 2.2. Exhibit a deduction of $(\neg p_2 \to p_1))$ from $\{(\neg p_1 \to p_2)$. Do not appeal to the deduction theorem.

Hint. First write out the deduction $\mathbf{D}$ of p_1 from $\{(\neg p_1 \to p_2)\,,\ \neg p_2\}$ that is implicitly given by the proof of part (a) of Lemma 2.3. Now use the proof of the Deduction Theorem to get the desired deduction. (The proof of the Deduction Theorem shows us how to put $\neg p_2 \to$ in front of all the lines of the given deduction and then to fix things up. There is one simplification here: If one puts $\neg p_2 \to$ in front of the formula $(\neg p_1 \to \neg p_2)$ that is on line 3 of $\mathbf{D}$, one gets an axiom. Thus one can forget about lines 1 and 2 of $\mathbf{D}$ and just begin with this axiom.)

Exercise 2.3. Show the following:

(a) $\vdash \neg(A \to B) \to \neg B$;

(b) $\vdash (A \vee \neg A)$.

A system $\mathbf{S}$ of deduction for $\mathcal{L}$ is *sound* if, for all sets Γ of formulas and all formulas A, if $\Gamma \vdash_{\mathbf{S}} A$ then $\Gamma \models A$.

An example of a system of deduction that is not sound can be gotten by adding to the axioms and rules for $\mathbf{SL}$ the extra axiom p_0. For this system $\mathbf{S}$, one has that $\emptyset \vdash_{\mathbf{S}} p_0$, but $\emptyset \not\models p_0$.

Theorem 2.4 (Soundness). *Let* Γ *be a set of formulas and let* A *be a formula. If* $\Gamma \vdash_{\mathbf{SL}} A$ *then* $\Gamma \models A$*. In other words,* **SL** *is sound.*

Proof. Let **D** be a deduction in **SL** of A from Γ. We shall show that, for every line C of **D**, $\Gamma \models C$. Applying this to the last line of **D**, this will give us that $\Gamma \models A$.

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

If $C \in \Gamma$ then trivially $\Gamma \models C$ (and so we have a contradiction).

It can easily be checked that all of our axioms are tautologies. If C is an axiom we have then that $\models C$ and so that $\Gamma \models C$.

Note that the rule of modus ponens is a *valid* rule, i.e., $\{D\,,\ (D \to E)\} \models E$ for any formulas D and E. Assume that C follows by MP from B and $(B \to C)$, where B and $(B \to C)$ are on earlier lines of **D**. Since C is the first "bad" line of **D**, $\Gamma \models B$ and $\Gamma \models (B \to C)$. By the validity of MP, it follows that $\Gamma \models C$. $\square$

A system **S** of deduction for $\mathcal{L}$ is *complete* if, for all sets Γ of formulas and all formulas A, if $\Gamma \models A$ then $\Gamma \vdash_{\mathbf{S}} A$.

Remark. Sometimes the word "complete" used to mean what we mean by "sound and complete."

We are now going to embark on the task of proving the completeness of **SL**. The proof will parallel the proof of the Compactness Theorem. In particular, the lemma that follows is the analogue of Lemma 1.6

Lemma 2.5. *Let* Γ *be a consistent (in* **SL***) set of formulas and let* A *be a formula. Then either* $\Gamma \cup \{A\}$ *is consistent or* $\Gamma \cup \{\neg A\}$ *is consistent.*

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

(i) $\Gamma \cup \{A\} \vdash B\,;$

(ii) $\Gamma \cup \{A\} \vdash \neg B\,;$

(iii) $\Gamma \cup \{\neg A\} \vdash B'\,;$

(iv) $\Gamma \cup \{\neg A\} \vdash \neg B'\,.$

Using Axiom Schema (4) together with (iii), (iv), and the Deduction Theorem, we can show that

$$\Gamma \vdash A\,.$$

This fact, together with (i) and (ii), allows us to show that $\Gamma \vdash B$ and $\Gamma \vdash \neg B$. Thus we have the contradiction that Γ is inconsistent. $\square$

Now we turn to the analogue of Lemma 1.7.

Lemma 2.6. *Let Γ be a consistent set of formulas. There is a set Γ^* of formulas such that*

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

(2) Γ^* *is consistent;*

(3) for every formula A, either A belongs to Γ^* or $\neg A$ belongs to Γ^*.

Proof. Let

$$A_0,\ A_1,\ A_2,\ A_3,\ \ldots$$

be the list (defined in the proof of Lemma 1.7) of all the formulas of $\mathcal{L}$. As in that proof we define, by recursion on natural numbers, a function that associates with each natural number n a set Γ_n of formulas.

Let $\Gamma_0 = \Gamma$.

Let

$$\Gamma_{n+1} \;=\; \begin{cases} \Gamma_n \cup \{A_n\} & \text{if } \Gamma_n \cup \{A_n\} \text{ is consistent;} \\ \Gamma_n \cup \{\neg A_n\} & \text{otherwise.} \end{cases}$$

Let $\Gamma^* = \bigcup_n \Gamma_n$.

Because $\Gamma = \Gamma_0 \subseteq \Gamma^*$, Γ^* has property (1).

Γ_0 is consistent. By Lemma 2.5, if Γ_n is consistent then so is Γ_{n+1}. By mathematical induction, every Γ_n is consistent. Suppose, in order to obtain a contradiction, that Γ^* is inconsistent. Let B be a formula such that $\Gamma^* \vdash B$ and $\Gamma^* \vdash \neg B$. Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be respectively deductions of B from Γ^* and of $\neg B$ from Γ^*. Let Δ be the set of all formulas belonging to Γ^* that are on lines of $\mathbf{D}_1$ or of $\mathbf{D}_2$. Then Δ is a finite subset of Γ^*, and so $\Delta \subseteq \Gamma_n$ for some n. But then $\Gamma_n \vdash B$ and $\Gamma_n \vdash \neg B$. This contradicts the consistency of Γ_n. Thus Γ^* has property (2).

Because either A_n or $\neg A_n$ belongs to Γ_{n+1} for each n and because each $\Gamma_{n+1} \subseteq \Gamma^*$, Γ^* has property (3). $\square$

Next comes the analogue of Lemma 1.8.

Lemma 2.7. *Let Γ^* be a set of formulas having properties (2) and (3) described in the statement of Lemma 2.6. Then Γ^* is satisfiable.*

Proof. Define a valuation v for $\mathcal{L}$ by setting

$$v(A) = \mathbf{T} \text{ if and only if } A \in \Gamma^*$$

for each sentence letter A. Let P be the property of being a formula A such that

$$v^*(A) = \mathbf{T} \text{ if and only if } A \in \Gamma^* .$$

We prove by formula induction that every formula has property P.

(i) For sentence letters, this is true by definition of v.

(ii) First we show that $\neg A \in \Gamma^*$ if and only if $A \notin \Gamma^*$ for any formula A. By (3) we have that $A \in \Gamma^*$ or $\neg A \in \Gamma^*$. If both A and $\neg A$ belong to Γ^*, then Γ^* is inconsistent, contrary to (2).

Now let A be a formula that has property P. Then

$$\begin{aligned} v^*(\neg A) = \mathbf{T} \quad & \text{if and only if} \quad v^*(A) = \mathbf{F} \\ & \text{if and only if} \quad A \notin \Gamma^* \\ & \text{if and only if} \quad \neg A \in \Gamma^* . \end{aligned}$$

(iii) We first show that $(A \vee B) \in \Gamma^*$ if and only if either $A \in \Gamma^*$ or $B \in \Gamma^*$, for any formulas A and B. Assume first that $(A \vee B) \in \Gamma^*$ but that $A \notin \Gamma^*$ and $B \notin \Gamma^*$. By (3), $\neg A \in \Gamma^*$ and $\neg B \in \Gamma^*$. Using the instance $(A \vee B) \to (\neg A \to B)$ of Axiom Schema (3) and two applications of MP, we see that $\Gamma^* \vdash B$. Since $\Gamma^* \vdash \neg B$, we get the contradiction that Γ^* is inconsistent. Next assume that $A \in \Gamma^*$ but $(A \vee B) \notin \Gamma^*$. By (3) $\neg(A \vee B) \in \Gamma^*$. Using the instance $A \to (A \vee B)$ of Axiom Schema (1), we again get the contradiction that Γ^* is inconsistent. The assumption that $B \in \Gamma^*$ but $(A \vee B) \notin \Gamma^*$ yields a similar contradiction with the aid of Axiom Schema (2).

Now let A and B be formulas that have property P. Then

$$\begin{aligned} v^*((A \vee B)) = \mathbf{T} \quad & \text{if and only if} \quad v^*(A) = \mathbf{T} \text{ or } v^*(B) = \mathbf{T} \\ & \text{if and only if} \quad A \in \Gamma^* \text{ or } B \in \Gamma^* \\ & \text{if and only if} \quad (A \vee B) \in \Gamma^* . \end{aligned}$$

Since, in particular, $v^*(A) = \mathbf{T}$ for every member of A of Γ^*, we have shown that Γ^* is satisfiable. $\square$

Theorem 2.8. *Let Γ be a consistent set of formulas. Then Γ is satisfiable.*

Proof. By Lemma 2.6, let Γ^* have properties (1)–(3) of that lemma. By Lemma 2.7, Γ^* is satisfiable. Hence Γ is satisfiable. $\square$

Theorem 2.9 (Completeness). *Let* Γ *be a set of formulas and let* A *be a formula such that* $\Gamma \models A$. *Then* $\Gamma \vdash_{\mathbf{SL}} A$. *In other words,* **SL** *is complete.*

Proof. Since $\Gamma \models A$, $\Gamma \cup \{\neg A\}$ is not satisfiable. By Theorem 2.8, $\Gamma \cup \{\neg A\}$ is inconsistent. Let B be a formula such that $\Gamma \cup \{\neg A\} \vdash B$ and $\Gamma \cup \{\neg A\} \vdash \neg B$. By the Deduction Theorem, $\Gamma \vdash (\neg A \rightarrow B)$ and $\Gamma \vdash \neg A \rightarrow \neg B)$. Using Axiom Schema 4, we can use these facts to show that $\Gamma \vdash A$. $\square$

Exercise 2.4. Derive Theorem 2.8 from Theorem 2.9.

Remark. Soundness and completeness imply compactness. To see this, assume that Γ is a set of formulas that is not satisfiable. By part (3) of Exercise 1.7, $\Gamma \models (p_0 \wedge \neg p_0)$. By completeness, $\Gamma \vdash (p_0 \wedge \neg p_0)$. Let $\mathbf{D}$ be a deduction of $(p_0 \wedge \neg p_0)$ from Γ. Let Δ be the set of all formulas $C \in \Gamma$ such that C is on a line of $\mathbf{D}$. Then Δ is a finite subset of Γ and $\Delta \vdash (p_0 \wedge \neg p_0)$. By soundness, $\Delta \models (p_0 \wedge \neg p_0)$. By part (3) of Exercise 1.7, Δ is not satisfiable. Thus Γ is not finitely satisfiable.

Exercise 2.5. Prove that $\{\neg(\neg A \wedge \neg B)\} \vdash (A \vee B)$ and that $\{(A \vee B)\} \vdash \neg(\neg A \wedge \neg B)$. You may use *any* of our theorems, lemmas, etc.

Exercise 2.6. We define by recursion on natural numbers a function that assigns to each natural number n a set $\mathbf{Formula}_n$ of formulas. Let $\mathbf{Formula}_0$ be the set of all sentence letters. Let A belong to $\mathbf{Formula}_{n+1}$ if and only if at least one of the following holds:

(i) $A \in \mathbf{Formula}_n$;

(ii) there is a $B \in \mathbf{Formula}_n$ such that A is $\neg B$;

(iii) there are $B \in \mathbf{Formula}_n$ and $C \in \mathbf{Formula}_n$ such that A is $(B \vee C)$.

It is not hard to prove that A is a formula if and only if A belongs to $\mathbf{Formula}_n$ for some n. (You may assume this.)

Use mathematical induction to prove that every formula has an even number of parentheses.

Exercise 2.7. Show, without using Completeness and Soundness, that $\vdash (\neg(\neg B \rightarrow A) \rightarrow \neg(A \vee B))$.

Exercise 2.8. Suppose we changed our system of deduction by replacing the Axiom Schemas 1 and 2 by the rules

$$\frac{A}{(A \vee B)} \qquad \frac{B}{(A \vee B)}$$

Would the resulting system be sound? Would it be complete?

Exercise 2.9. Show, without using completeness and soundness, that

$$\{(A \to C)\,,\ (B \to C)\} \ \vdash \ ((A \vee B) \to C)\,.$$

Exercise 2.10. Use the Deduction Theorem and its converse (and none of our other results) to give a brief proof that $\vdash (B \to (A \to A))$.

3 The semantics of pure first-order predicate logic

We now begin our study of what is called, among other things, *predicate logic*, *quantificational logic*, and *first-order logic*. We shall use the term "first-order logic" for our subject. The term "predicate logic" suggests formal languages that have predicate lettrers but not function letters, and we do not want to leave out the latter. Both "predicate logic" and "quantificational logic" fail to suggest that higher-order and infinitary logics are excluded, and—except for a brief consideration of second-order logic at the end of the course—we do intend to exclude them.

In order that our first pass through first-order logic be as free of complexities as possible, we study in this section a simplified version of first-order logic, one whose formal languages lack two important kinds of symbols:

(a) function letters;

(b) an identity symbol.

We call this simplified logic "*pure* first-order *predicate* logic." In the next section, we shall see what changes have to be made in our definitions and proofs to accommodate the presence of these symbols.

The languages $\mathcal{L}^*_{\mathrm{C}}$ of pure predicate logic.

For each any set C of *constant symbols*, we shall have a language $\mathcal{L}^*_{\mathrm{C}}$.

Symbols of $\mathcal{L}^*_{\mathrm{C}}$:

(i)	sentence letters	$p_0, p_1, p_2, \ldots$
(ii)	for each $n \geq 1$, n-place predicate letters	$P^n_0, P^n_1, P^n_2, \ldots$
(iii)	variables	$v_0, v_1, v_2, \ldots$
(iv)	constant symbols (constants)	all members of C
(v)	connectives	$\neg$, $\vee$
(vi)	quantifier	$\forall$
(vii)	parentheses	(,)

Constants and variables will more or less play the role played in natural languages by nouns and pronouns respectively. Predicate letters will more or less play the role that predicates play in natural languages. In combination, these symbols will give our formal language a new kind of basic formulas, the simplest of which will play the role that subject-predicate sentences play in natural languages. The quantifier $\forall$ will play the role that the phrase "for all" can play in natural languages.

Formulas of $\mathcal{L}^*_{\mathrm{C}}$

(1) Each sentence letter is a formula.

(2) For each n and i, if $t_1, \ldots, t_n$ are variables or constants, then $P^n_i t_1 \ldots t_n$ is a formula.

(3) If A is a formula, then so is $\neg A$.

(4) If A and B are formulas, then so is $(A \vee B)$.

(5) If A is a formula and x is a variable, then $\forall x A$ is a formula.

(6) Nothing is a formula unless its being one follows from (1)–(5).

The formulas given by (1) and (2) are called *atomic* formulas.

The method of proof by formula induction applies to $\mathcal{L}^*_{\mathrm{C}}$ as it does to $\mathcal{L}$. To prove by formula induction that every formula of $\mathcal{L}^*_{\mathrm{C}}$ has property P, we must prove (i), (ii), (iii), and (iv) below.

(i) Each atomic formula has property P.

(ii) If A is a formula that has property P, then $\neg A$ has P.

(iii) If A and B are formulas that have property P, then $(A \vee B)$ has P.

(iv) If x is a variable and A is a formula that has property P, then $\forall x A$ has P.

Not only is there a step, step (iv), that was absent in the case of $\mathcal{L}$, but also there is an extra part to step (ii), the part corresponding to atomic formulas of the form $P^n_i t_1 \ldots t_n$.

Unique readability holds for $\mathcal{L}^*_{\mathrm{C}}$ as it does for $\mathcal{L}$. Here are the new versions of the Lemmas 1.1–1.4 that were used to prove the unique readability theorem, Theorem 1.5. The proofs are similar to the proofs of the earlier lemmas and theorem.

Lemma 3.1. *Every formula of $\mathcal{L}^*_{\mathrm{C}}$ contains the same number of (occurrences of) left parentheses as (occurrences of) right parentheses.*

Lemma 3.2. *For every formula A of $\mathcal{L}^*_{\mathrm{C}}$,*

(a) every initial segment of A has the at least as many left as right parentheses;

(b) if A is a disjunction (i.e., if A is $(B \vee C)$ for some B and C), then every proper initial segment of A (i.e., every initial segment of A that is not the whole of A) that has length greater than 0 has more left than right parentheses.

Lemma 3.3. *For every formula A of $\mathcal{L}_{\mathrm{C}}^*$, exactly one of the following holds.*

(1) A is an atomic formula.

(2) There is a formula B such that A is $\neg B$.

(3) There are formulas B and C such that A is $(B \vee C)$.

(4) There is a formula B and there is a variable x such that A is $\forall x B$.

Lemma 3.4. *No proper initial segment of a formula of $\mathcal{L}_{\mathrm{C}}^*$is a formula of $\mathcal{L}_{\mathrm{C}}^*$.*

Theorem 3.5 (Unique Readability). *Let A be a formula of $\mathcal{L}_{\mathrm{C}}^*$. Then exactly one of the following holds.*

(1) A is an atomic formula.

(2) There is a unique formula B such that A is $\neg B$.

(3) There are unique formulas B and C such that A is $(B \vee C)$.

(4) There is a unique formula B and there is a unique variable x such that A is $\forall x B$.

Remark. Note that we could have phrased Lemma 1.3 exactly as Lemma 3.2 is phrased without altering its content in any significant way.

Truth and logical implication.

As we did with the sentential language $\mathcal{L}$, we want to introduce semantic notions for the languages $\mathcal{L}_{\mathrm{C}}^*$. If we want to keep as close as possible to the methods of §1, then we might try to extend the notion of a valuation v so that v assigns a truth-value to all atomic formulas, not just all sentence letters. But consider an atomic formula like $P_1^2 v_3 c$. The symbol v_3 is a *variable*, i.e., we are not going to use it to denote any particular object. The *language* of arithmetic does not provide a truth-value for an expression like "$x < 3$," and the English *language* does not provide a truth-value for sentences like "He is fat." To get a truth-value for the former, one needs to assign the variable x to some particular number. To get a truth-value for the latter, one needs a context in which "he" denotes a particular person (or animal or whatever). Similarly, the semantics of $\mathcal{L}^*_{\mathrm{C}}$ will not by itself provide a truth-value for $P^2_1 v_3 c$. In addition, there will have to be an assignment of v_3 to some particular object.

What are the objects over which our variables are to range? A natural answer would be that they range over *all* objects. If we made this choice, then we could interpret $\forall v_3$ as saying "for all objects v_3." However, there are reasons for not wanting to make a matter of *logic* that, e.g., there are more than 17 objects, and requiring that our variables range over all objects would make this a matter of logic. Therefore we allow the variables to range over any set of objects, and we make the specification of such a set part of any interpretatation of our language.

The first step in providing an interpretation of $\mathcal{L}^*_{\mathrm{C}}$ (or, as we shall say, a *model for* $\mathcal{L}^*_{\mathrm{C}}$) is thus to specify a set $\boldsymbol{D}$ as the *domain* or *universe* of the model. It is standard to require that $\boldsymbol{D}$ be a *non-empty* set, because doing so avoids certain technical complexities. We make this requirement.

The second step is to provide a way to assign truth-values to atomic formulas when their variables are assigned to particular members of $\boldsymbol{D}$. To accomplish this (in an indirect way), (i) we specify the truth-values of sentence letters and (ii) we specify what property of elements of $\boldsymbol{D}$ or relation among elements of $\boldsymbol{D}$ each predicate letter is to stand for. We do this by telling, for each n and i, which n-tuples $(d_1, \ldots, d_n)$ of elements of $\boldsymbol{D}$ the predicate P^n_i is true of.

The final step in determining a model for $\mathcal{L}^*_{\mathrm{C}}$ is to specify what element of $\boldsymbol{D}$ each constant denotes.

Here is the formal definition. A *model* for $\mathcal{L}^*_{\mathrm{C}}$ is a triple $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$, where

(i) $\boldsymbol{D}$ is a non-empty set (the *domain* or *universe* of $\mathfrak{M}$);

(ii) v is a function (the *valuation* of $\mathfrak{M}$) that assigns a truth-value to each sentence letter and each $(n+1)$-tuple of the form $(P^n_i, d_1, \ldots, d_n)$ for $d_1, \ldots, d_n$ members of $\boldsymbol{D}$.

(iii) χ is a function (the *constant assignment* of $\mathfrak{M}$) that assigns to each constant an element of $\boldsymbol{D}$.

Note that, except for the sentence letters, the things to which v assigns truth-values are not actually formulas.

In describing the v of a model, we shall often find it convenient to list the set of things to which v assigns $\mathbf{T}$. Let us call this the *v-truth set*.

Examples:

(a) Let $\mathrm{C}_a = \{c\}$. Let $\mathfrak{M}_a = (\boldsymbol{D}_a, v_a, \chi_a)$, where:

$$\begin{array}{rcl} \boldsymbol{D}_a & = & \{d_1, d_2\} \\ v_a\text{-truth set} & = & \{p_2, (P^1, d_1), (P^2, d_1, d_2), (P^2, d_2, d_2)\} \\ \chi_a(c) & = & d_2 \end{array}$$

(b) Let $\mathrm{C}_b = \{c, c'\}$. Let $\mathfrak{M}_b = (\boldsymbol{D}_b, v_b, \chi_b)$, where:

$$\begin{array}{rcl} \boldsymbol{D}_b & = & \{0, 1, 2, \ldots\} \\ v_b\text{-truth set} & = & \{(P^1, 0)\} \cup \{(P^2, m, n) \mid m \geq n\} \\ \chi_b(c) & = & 0 \\ \chi_b(c') & = & 1 \end{array}$$

Whenever we omit the subscript of a predicate letter, as we have done in describing these two models, let us take the omitted subscript to be 0.

Let $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$ be a model for $\mathcal{L}^*_{\mathrm{C}}$. Let s be a *variable assignment*, a function that assigns a member $s(x)$ of $\boldsymbol{D}$ to each variable x. For each variable or constant t, let

$$\mathrm{den}^s_{\mathfrak{M}}(t) = \begin{cases} s(t) & \text{if } t \text{ is a variable;} \\ \chi(t) & \text{if } t \text{ is a constant.} \end{cases}$$

By recursion on formulas, we define a function $v^s_{\mathfrak{M}}$ that assigns a truth-value to each formula.

(i) The case of A atomic:

(a) $v^s_{\mathfrak{M}}(p_i) = v(p_i)$;
(b) $v^s_{\mathfrak{M}}(P^n_i t_1 \ldots t_n) = v((P^n_i, \mathrm{den}^s_{\mathfrak{M}}(t_1), \ldots, \mathrm{den}^s_{\mathfrak{M}}(t_n)))$;

(ii) $v^s_{\mathfrak{M}}(\neg A) = \begin{cases} \mathbf{F} & \text{if } v^s_{\mathfrak{M}}(A) = \mathbf{T}; \\ \mathbf{T} & \text{if } v^s_{\mathfrak{M}}(A) = \mathbf{F}; \end{cases}$

(iii) $v^s_{\mathfrak{M}}((A \vee B)) = \begin{cases} \mathbf{T} & \text{if } v^s_{\mathfrak{M}}(A) = \mathbf{T} \text{ and } v^s_{\mathfrak{M}}(B) = \mathbf{T}; \\ \mathbf{T} & \text{if } v^s_{\mathfrak{M}}(A) = \mathbf{T} \text{ and } v^s_{\mathfrak{M}}(B) = \mathbf{F}; \\ \mathbf{T} & \text{if } v^s_{\mathfrak{M}}(A) = \mathbf{F} \text{ and } v^s_{\mathfrak{M}}(B) = \mathbf{T}; \\ \mathbf{F} & \text{if } v^s_{\mathfrak{M}}(A) = \mathbf{F} \text{ and } v^s_{\mathfrak{M}}(B) = \mathbf{F}; \end{cases}$

$$(\text{iv})\ \ v^s_{\mathfrak{M}}(\forall x A) \ = \begin{cases} \mathbf{T} & \text{if for all } d \in \boldsymbol{D},\ v^{s'}_{\mathfrak{M}}(A) = \mathbf{T}, \\ & \text{where } s' \text{ is like } s \text{ except that } s'(x) = d; \\ \mathbf{F} & \text{otherwise}. \end{cases}$$

$\mathfrak{M}$ *satisfies* A *under* s if and only if $v^s_{\mathfrak{M}}(A) = \mathbf{T}$.

An occurrence of a variable x in a formula A is *free* if the occurrence is not within any subformula of A of the form $\forall x B$. A *sentence* or *closed formula* is a formula with no free occurrences of variables.

Example. The third occurrence of v_1 in the formula

$$\forall v_2(\forall v_1 P^1_3 v_1 \ \vee \ P^2_1 v_1 v_2)$$

is free, and so this formula is not a sentence.

It is not hard to verify that whether or not $\mathfrak{M}$ satisfies A under s does not depend on the whole of s but only on the values $s(x)$ for variables x that have free occurrences in A. For sentences A, we may then define $v_{\mathfrak{M}}(A)$ to be the common value of all $v^s_{\mathfrak{M}}(A)$. We define a sentence A to be *true* in $\mathfrak{M}$ if $v_{\mathfrak{M}}(A) = \mathbf{T}$ and *false* in $\mathfrak{M}$ if $v_{\mathfrak{M}}(A) = \mathbf{F}$. $\mathfrak{M}$ *satisfies* a set Γ of formulas under s if and only if all $\mathfrak{M}$ satisfies each member of Γ under s. A set of sentences is *true* in $\mathfrak{M}$ if and only if all its members are true in $\mathfrak{M}$.

We introduce one more abbreviation:

$$\exists x A \quad \text{for} \quad \neg\forall x \neg A.$$

It is not hard to verify that the defined symbol $\exists$ obeys the following rule:

$$(\text{v})\ \ v^s_{\mathfrak{M}}(\exists x A) \ = \begin{cases} \mathbf{T} & \text{if for some } d \in \boldsymbol{D},\ v^{s'}_{\mathfrak{M}}(A) = \mathbf{T}, \\ & \text{where } s' \text{ is like } s \text{ except that } s'(x) = d; \\ \mathbf{F} & \text{otherwise}. \end{cases}$$

Example. Here are some sentences true in the model $\mathfrak{M}_a$ described on page 27: $\neg P^1 c$; $\forall v_1 \exists v_2\, P^2 v_1 v_2$; $\exists v_1(p_2 \wedge P^2 v_1 v_1)$.

Exercise 3.1. For each of the following sentences, tell in which of the models $\mathfrak{M}_a$ and $\mathfrak{M}_b$ the sentence is true. Explain your answers briefly and informally.

(a) $\exists v_1 \forall v_2\, P^2 v_2 v_1$ $\quad$ (b) $\forall v_1(P^1 v_1 \vee P^2 c v_1)$
(c) $\forall v_1(P^1 v_1 \to p_2)$ $\quad$ (d) $\exists v_1(P^1 v_1 \to p_2)$

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

$$\text{if } \mathfrak{M} \text{ satisfies } \Gamma \text{ under } s, \text{ then } \mathfrak{M} \text{ satisfies } A \text{ under } s\,.$$

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

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

Exercise 3.2. For each of the following pairs (Γ, A), tell whether $\Gamma \models A$. 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: $\{\forall v_1 \exists v_2\, P^2 v_1 v_2\}$; A: $\exists v_2 \forall v_1\, P^2 v_1 v_2$.

(b) Γ: $\{\exists v_1 \forall v_2\, P^2 v_1 v_2\}$; A: $\forall v_2 \exists v_1\, P^2 v_1 v_2$.

(c) Γ: $\{\forall v_1\;\; P^2 v_1 v_1, P^2 c_1 c_2\}$; A: $P^2 c_2 c_1$;

(d) Γ: $\{\forall v_1 \forall v_2\, P^2 v_1 v_2\}$; A: $\forall v_2 \forall v_1\, P^2 v_1 v_2$;

(e) Γ: $\{P^1 v_1\}$; A: $\forall v_1\, P^1 v_1$.

Exercise 3.3. Describe a model in which the following sentences are all true.

(a) $\forall v_1 \exists v_2\, P^2 v_1 v_2$.

(b) $\forall v_1 \forall v_2 (P^2 v_1 v_2 \to \neg P^2 v_2 v_1)$.

(c) $\forall v_1 \forall v_2 \forall v_3 ((P^2 v_1 v_2 \wedge P^2 v_2 v_3) \to P^2 v_1 v_3)$.

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

Exercise 3.4. Show that the four statements of Exercise 1.7 hold for formulas of $\mathcal{L}^*_{\mathrm{C}}$.

For sentential logic, valid formulas and tautologies are by definition the same. For predicate logic, the notion of a tautology is different from that of a valid formula. We now explain how this difference arises.

Call a formula of any of our formal languages *sententially atomic* if it is neither a negation nor a disjuction, i.e., if it is not $\neg A$ for any A and it is not $(A \vee B)$ for any A and B. The sententially atomic formulas of $\mathcal{L}$ are the sentence letters. The sententially atomic formulas of $\mathcal{L}_{\mathrm{C}}^{*}$ are the atomic formulas and the quantifications (the formulas of the form $\forall x A$). Note that every formula can be gotten from the sententially atomic formulas using only negation and conjunction.

An *extended valuation* for any of our languages is a function that assigns a truth-value to each sententially atomic formula.

Let v be an extended valuation. We as follows define a function v^* that assigns a truth-value to each formula.

(a) if A is sententially atomic, then $v^*(A) = v(A)$;

(b) $v^*(\neg A) = \begin{cases} \mathbf{F} & \text{if } v^*(A) = \mathbf{T}; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F}; \end{cases}$

(c) $v^*((A \vee B)) = \begin{cases} \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{T}; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{F}; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{T}; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{F}. \end{cases}$

We define a formula A to be *true* under the extended valuation v if $v^*(A) = \mathbf{T}$ and to be *false* under v if $v^*(A) = \mathbf{F}$. We define a set Γ of formulas to be *true* under v if and only if all members of Γ are true under v.

If Γ is a set of formulas and A is a formula, then say that Γ *sententially implies* A if and only if A is true under every extended valuation under which Γ is true. We write $\Gamma \models_{\mathrm{sl}} A$ to mean that Γ sententially implies A. A formula A is a *tautology* if and only if $\emptyset \models_{\mathrm{sl}} A$, i.e., if and only if A is true under every extended valuation. We usually write $\models_{\mathrm{sl}} A$ instead of $\emptyset \models_{\mathrm{sl}} A$.

For the language $\mathcal{L}$, the new definition of tautology agrees with the old definition. It is easy to see that for both $\mathcal{L}$ and $\mathcal{L}_{\mathrm{C}}^{*}$ every tautology is valid. The converse, while true for $\mathcal{L}$, is false for $\mathcal{L}_{\mathrm{C}}^{*}$. For example, the formula

$$\forall v_1 (P^1 v_1 \vee \neg P^1 v_1)$$

is valid but is not a tautology.

We now begin the proof of the Compactness Theorem for $\mathcal{L}^*_{\mathrm{C}}$. As we did with $\mathcal{L}$, we call a set Γ of formulas of $\mathcal{L}^*_{\mathrm{C}}$ *finitely satisfiable* if every finite subset of Γ is satisfiable. The Compactness Theorem we shall prove states that every finitely satisfiable set of sentences is satisfiable. The stronger statement with "sentences" replaced by "formulas" is true. The reasons why we prove only the weaker one are (a) simplicity and (b) considerations—to be explained later—involving the theory of deduction.

The analogue for formulas of the following lemma is true and has a proof like that of the lemma.

Lemma 3.6. *Let Γ be a finitely satisfiable set of sentences of $\mathcal{L}^*_{\mathrm{C}}$ and let A be a sentence of $\mathcal{L}^*_{\mathrm{C}}$. Then either $\Gamma \cup \{A\}$ is finitely satisfiable or $\Gamma \cup \{\neg A\}$ is finitely satisfiable.*

Proof. The proof is like the proof of Lemma 1.6. Assume for a contradiction neither $\Gamma \cup \{A\}$ nor $\Gamma \cup \{\neg A\}$ is finitely satisfiable. It follows that there are finite subsets Δ and Δ' of Γ such that neither $\Delta \cup \{A\}$ nor $\Delta' \cup \{\neg A\}$ is satisfiable. Since Γ is finitely satisfiable, the finite subset $\Delta \cup \Delta'$ of Γ is satisfiable. Let $\mathfrak{M}$ be a model in which $\Delta \cup \Delta'$ is true. If $v_{\mathfrak{M}}(A) = \mathbf{T}$, then $\Delta \cup \{A\}$ is true in $\mathfrak{M}$ and so $\Delta \cup \{A\}$ is satisfiable. Otherwise $\Delta' \cup \{\neg A\}$ is true in $\mathfrak{M}$ and so $\Delta' \cup \{\neg A\}$ is satisfiable. In either case we have a contradiction. $\square$

Simplifying assumption. From now on we assume that the members of the set C can be arranged in a finite or infinite list. In the technical jargon of set theory, this is the assumption that C is *countable*. Most of the facts we shall prove can be proved without this assumption, but the proofs involve concepts beyond the scope of this course.

Our next lemma is the analogue for $\mathcal{L}^*_{\mathrm{C}}$ of Lemma 1.7. The main difference from the earlier lemma is that the set Γ^* has a fourth property. This property will be needed for the proof of Lemma 3.8, the analogue of Lemma 1.8. If A is a formula, x is a variable, and t is a variable or constant, then $A(x;t)$ is the result of replacing each free occurrence of x in A by an occurrence of t. A set Γ of formulas is *Henkin* if and only if, for each formula A and each variable x, if (i) below holds, then (ii) also holds.

(i) $A(x;c) \in \Gamma$ for all $c \in \mathrm{C}$.

(ii) $\forall x A \in \Gamma$.

Lemma 3.7. *Let Γ be a finitely satisfiable set of sentences of $\mathcal{L}^*_{\mathrm{C}}$. Let C^* be a set gotten from C by adding infinitely many new constants. There is a set Γ^* of sentences of $\mathcal{L}^*_{\mathrm{C}^*}$ such that*

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

(2) Γ^* *is finitely satifiable;*

(3) for every sentence A of $\mathcal{L}^*_{\mathrm{C}^*}$, either A belongs to Γ^* or $\neg A$ belongs to Γ^*;

(4) Γ^* *is Henkin.*

Proof. In keeping with our simplying assumption, let $c_0, c_1, \ldots,$ be all the constants of $\mathcal{L}^*_{\mathrm{C}^*}$.

Since $\mathcal{L}^*_{\mathrm{C}^*}$ has symbols that are not symbols of $\mathcal{L}$, we need to specify an alphabetical order for the symbols of $\mathcal{L}^*_{\mathrm{C}^*}$. Let that order be as follows.

$$\begin{array}{l} \neg,\ \vee,\ (,\ ),\ \forall, \\ v_0,\ v_1,\ v_2, \ldots, \\ c_0,\ c_1,\ c_2, \ldots, \\ p_0,\ p_1,\ p_2, \ldots, \\ P^1_0, P^1_1, P^1_2, \ldots, \\ P^2_0, P^2_1, P^2_2, \ldots, \\ \ldots . \end{array}$$

Now we give a method for listing all the sentences of $\mathcal{L}^*_{\mathrm{C}^*}$ in an infinite list. First list in alphabetical order all the sentences that have length 1 and contain no occurrences of variables, constants, sentence letters, or predicate letters with any subscript or superscript larger than 0. Next list in alphabetical order all the sentences that have length ≤ 2 and contain no occurrences of variables, constants, sentence letters, or predicate letters with any subscript or superscript larger than 1. Next list in alphabetical order all the sentences that have length ≤ 3 and contain no occurrences of variables, constants, sentence letters, or predicate letters with any subscript or superscript larger than 2. Continue in this way.

Let the sentences of $\mathcal{L}^*_{\mathrm{C}^*}$, in the order listed, be

$$A_0,\ A_1,\ A_2,\ A_3,\ \ldots .$$

We define, by recursion on natural numbers, a function that associates with each natural number n a set Γ_n of sentences of $\mathcal{L}^*_{\mathrm{C}}$.

Let $\Gamma_0 = \Gamma$.

For each n, we shall make sure that at most two sentences belong to Γ_{n+1} but not to Γ_n. Since none of the constants added to C to get C* occur in sentences in Γ, it follows that for each n only finitely many of the new constants occur in sentences in Γ_n.

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

$$\Gamma'_n = \begin{cases} \Gamma_n \cup \{A_n\} & \text{if } \Gamma_n \cup \{A_n\} \text{ is finitely satisfiable;} \\ \Gamma_n \cup \{\neg A_n\} & \text{otherwise.} \end{cases}$$

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

(a) $\neg A_n \in \Gamma'_n$.

(b) A_n is $\forall x_n B_n$ for some variable x_n and formula B_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 Γ'_n. Such an i must exist, since only finitely many of the infinitely many new constants occur in sentences in Γ'_n. Let

$$\Gamma_{n+1} = \Gamma'_n \cup \{\neg B_n(x_n; c_{i_n})\}.$$

Let $\Gamma^* = \bigcup_n \Gamma_n$.

Because $\Gamma = \Gamma_0 \subseteq \Gamma^*$, Γ^* has property (1).

We prove by mathematical induction that Γ_n is finitely satisfiable for each n. Γ_0 is finitely satifiable by hypothesis.[1] Assume that Γ_n is finitely satisfiable. Lemma 3.6 implies that Γ'_n is finitely satisfiable. If $\Gamma_{n+1} = \Gamma'_n$, then Γ_{n+1} is finitely satisfiable. Assume then that $\Gamma_{n+1} = \Gamma'_n \cup \{\neg B_n(x_n; c_{i_n})\}$ and, in order to derive a contradiction, assume that Γ_{n+1} is not finitely satisfiable. For some finite subset Δ of Γ'_n, $\Delta \cup \{\neg B_n(x_n; c_{i_n})\}$ is not satisfiable. Since Γ'_n is finitely satisfiable and $\neg A_n \in \Gamma'_n$, $\Delta \cup \{\neg A_n\}$ is satisfiable. Let $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$ be a model for $\mathcal{L}^*_{\mathrm{C}^*}$ in which $\Delta \cup \{\neg A_n\}$ is true. Since A_n is $\forall x_n B_n$, $\forall x_n B_n$ is false in $\mathfrak{M}$. This means that there is a $d \in \boldsymbol{D}$ such that $v^s_{\mathfrak{M}}(B_n) = \mathbf{F}$ for any variable assignment s such that $s(x_n) = d$. Let $\mathfrak{M}' = (\boldsymbol{D}, v, \chi')$ be just like $\mathfrak{M}$, except let

$$\chi(c_{i_n}) = d.$$

[1] Actually there is a subtlety here. The assumption that Γ is finitely satisfiable, if precisely formulated, says that every finite subset of Γ is true in some model for $\mathcal{L}^*_{\mathrm{C}}$. But we want Γ_0 to be finitely satisfiable in the sense that every finite subset of Γ_0 is true in a model for $\mathcal{L}^*_{\mathrm{C}^*}$. Nevertheless, there is no problem. Any model for $\mathcal{L}^*_{\mathrm{C}}$ can be made into a model for $\mathcal{L}^*_{\mathrm{C}^*}$ by defining χ of the new constants in an arbitrary way. Since Γ_0 contains none of the new constants, subsets of it will be true in the resulting model if and only if they are true in the given one.

Since c_{i_n} does not occur in B_n,

$$v_{\mathfrak{M}'}(B_n(x_n; c_{i_n})) = \mathbf{F}.$$

Since c_{i_n} does not occur in Δ, Δ is true in $\mathfrak{M}'$. Thus we have the contradiction that $\Delta \cup \{\neg B_n(x_n; c_{i_n})\}$ is satisfiable.

If Δ is any finite subset of Γ^*, then $\Delta \subseteq \Gamma_n$ for some n. Since Γ_n is finitely satisfiable, Δ is satisfiable. Thus Γ^* has property (2).

Because either A_n or $\neg A_n$ belongs to Γ_{n+1} for each n and because each $\Gamma_{n+1} \subseteq \Gamma^*$, Γ^* has property (3).

Suppose that A_n is $\forall x_n B_n$. If $A_n \notin \Gamma^*$, then $A_n \notin \Gamma_{n+1}$ and so $\neg A_n \in \Gamma_{n+1}$. But this implies that $\neg B_n(x_n; c_{i_n}) \in \Gamma_{n+1} \subseteq \Gamma^*$. By property (2) of Γ^*, it follows that $B_n(x_n; c_{i_n}) \notin \Gamma^*$. This argument shows that Γ^* has property (4). $\square$

Lemma 3.8. *Let Γ^* be a set of sentences of a language $\mathcal{L}^*_{\mathrm{C}^*}$ having properties (2), (3), and (4) described in the statement of Lemma 3.7. Then Γ^* is satisfiable.*

Proof. Define a model $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$ for $\mathcal{L}^*_{\mathrm{C}^*}$ as follows.

(i) $\boldsymbol{D} = \mathrm{C}^*$.

(ii) (a) $v(p_i) = \mathbf{T}$ if and only if $p_i \in \Gamma^*$.

(b) $v((P^n_i, c_1, \ldots, c_n)) = \mathbf{T}$ if and only if $P^n_i c_1 \ldots c_n \in \Gamma^*$.

(iii) $\chi(c) = c$ for each $c \in \mathrm{C}^*$.

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

$$v_{\mathfrak{M}}(A) = \mathbf{T} \text{ if and only if } A \in \Gamma^*.$$

We prove, by a variant of formula induction, that every sentence of $\mathcal{L}^*_{\mathrm{C}^*}$ has property P.

(i)(a) Sentence letters have P because $v_{\mathfrak{M}}(p_i) = v(p_i)$.

(i)(b) Atomic sentences $P^n_i c_1 \ldots c_n$ have P because

$$v_{\mathfrak{M}}(P^n_i c_1 \ldots c_n) = v((P^n_i, \chi(c_1), \ldots, \chi(c_n))) = v((P^n_i, c_1, \ldots, c_n)).$$

(ii) and (iii) The proof that $\neg A$ has P if A has P and that $(A \vee B)$ has P if A and B have P are just like the corresponding steps of the proof of Lemma 1.8.

(iv) Let A be a formula with no free occurrences of variables other than the variable x. Assume that, for every $c \in \mathrm{C}^*$, $A(x;c)$ has P. We prove that $\forall x A$ has P.

$$\begin{aligned} v_{\mathfrak{M}}(\forall x A) = \mathbf{T} \quad &\text{iff} \quad \text{for all } s, v^s_{\mathfrak{M}}(A) = \mathbf{T} \\ &\text{iff} \quad \text{for all } c \in \mathrm{C}^*, \text{ for all } s \text{ with } s(x) = c,\ v^s_{\mathfrak{M}}(A) = \mathbf{T} \\ &\text{iff} \quad \text{for all } c \in \mathrm{C}^*,\ v_{\mathfrak{M}}(A(x;c)) = \mathbf{T} \\ &\text{iff} \quad \text{for all } c \in \mathrm{C}^*,\ A(x;c) \in \Gamma^* \\ &\text{iff} \quad \forall x A \in \Gamma^* \end{aligned}$$

The first "iff" is by the definition of $v_{\mathfrak{M}}$ and the fact that no variable besides x occurs free in A. The second "iff" is by the fact that no variable besides x occurs free in A and the fact that $\boldsymbol{D} = \mathrm{C}^*$. The third "iff" is by the fact that $\chi(c) = c$ for each $c \in \mathrm{C}^*$. The fourth "iff" is by the fact that the sentences $A(x;c)$ have property P.

To see that the "if" part of the last "iff" holds, assume that $\forall x A \in \Gamma^*$ and that, for some $c \in \mathrm{C}^*$, $A(x;c) \notin \Gamma^*$. By (3), $\neg A(x;c) \in \Gamma^*$. Thus $\{\forall x A, \neg A(x;c)\}$ is a finite subset of Γ^*. But this subset is not satisfiable, contradicting (2).

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

Since, in particular, $v_{\mathfrak{M}}(A) = \mathbf{T}$ for every member of A of Γ^*, we have shown that Γ^* is satisfiable. □

Theorem 3.9 (Compactness). *Let Γ be a finitely satisfiable set of sentences of $\mathcal{L}^*_{\mathrm{C}}$. Then Γ is satisfiable, i.e., true in a model for $\mathcal{L}^*_{\mathrm{C}}$.*

Proof. By Lemma 3.7, let Γ^* have properties (1)–(4) of that lemma. By Lemma 3.8, Γ^* is satisfiable. Let $\mathfrak{M}^*$ be a model for $\mathcal{L}^*_{\mathrm{C}^*}$ in which Γ^* is true. By (1), Γ is true in $\mathfrak{M}^*$. We can turn $\mathfrak{M}^*$ into a model for $\mathcal{L}^*_{\mathrm{C}}$ in which Γ is true by throwing away the part of the χ of Γ^* that assigns objects to the constants in C^* that do not belong to C. (The resulting model $\mathfrak{M}$ is called the *reduct* of $\mathfrak{M}^*$ to $\mathcal{L}^*_{\mathrm{C}}$.) □

Corollary 3.10 (Compactness, Second Form). *Let Γ be a set of sentences of $\mathcal{L}^*_{\mathrm{C}}$ and let A be a sentence such that $\Gamma \models A$. Then there is a finite subset Δ of Γ such that $\Delta \models A$.*

Proof. The proof is just like that of Corollary 1.10. □

Exercise 3.5. For each of the following pairs (Γ, A), tell whether $\Gamma \models_{\text{sl}} A$. Prove your answers.

(a) Γ: $\{\forall v_1\, P^1 v_1,\ \forall v_1(P^1 v_1 \to P^1 v_2\}$; A: $P^1 v_2$;

(b) Γ: $\{(\forall v_1\, \neg P^1 v_1 \ \to\ p_0),\ (\neg\forall v_2\, P^1 v_2 \ \to\ \neg p_0)\}$; A: $(\forall v_2 P^1 v_2 \vee \exists v_1\, P^1 v_1)$:

Exercise 3.6. Let Γ^* be a set of sentences having properties (2) and (3) described in the statement of Lemma 3.7. Show that Γ^* is Henkin if and only if, for each formula A and each variable x, if (iii) below holds, then (iv) also holds.

(iii) $\exists x A \in \Gamma^*$.

(iv) $A(x; c) \in \Gamma^*$ for some $c \in \mathrm{C}^*$.

Exercise 3.7. Let $\mathrm{C} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$, where, e.g., $\mathbf{7}$ is the numeral "7." Let $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$, where:

$$
\begin{aligned}
\boldsymbol{D} &= \{0, 1, 2, \ldots\} \\
v\text{-truth set} &= \{(P^2, m, n) \mid m \geq n\} \cup \{(P_0^3, m, n, p) \mid m + n = p\} \\
&\qquad \cup \{(P_1^3, m, n, p) \mid m \cdot n = p\} \\
\chi(\mathbf{n}) &= n
\end{aligned}
$$

Let Σ be the set of all sentences true in $\mathfrak{M}$. Prove that there is a model $\mathfrak{M}' = (\boldsymbol{D}', v', \chi')$ such that:

(a) Σ is true in $\mathfrak{M}'$;

(b) there is a $d \in \boldsymbol{D}'$ such that, for every natural number n, $(P^2, d, \chi'(\mathbf{n}))$ belongs to the v'-truth set.

Hint. Let $\mathrm{C}^* = \mathrm{C} \cup \{c\}$. Describe the set Π of sentences involving c that need to be true in $\mathfrak{M}'$ if $\chi'(c)$ is to be a d witnessing that (b) holds. Next show that $\Sigma \cup \Pi$ is finitely satisfiable. To do this, assume that Δ is a finite subset of Π. Show that $\mathfrak{M}$ can be made into a model $\bar{\mathfrak{M}} = (\boldsymbol{D}, v, \bar{\chi})$ of $\Sigma \cup \Delta$ by an appropriate choice of $\bar{\chi}(c)$. Apply the Compactness Theorem to get a model $\mathfrak{M}^*$ of $\Sigma \cup \Pi$. Finally let $\mathfrak{M}'$ be the reduct of $\mathfrak{M}^*$ to $\mathcal{L}^*_{\mathrm{C}}$.

4 The semantics of full first-order logic

In this section we make two additions to the languages $\mathcal{L}^*_{\mathrm{C}}$ of §3. The first is the addition of a symbol for identity. The second is the addition of symbols that are used to denote functions.

The languages $\mathcal{L}^*_{=,\mathrm{C}}$ of predicate logic with identity.

For each set C of constant symbols, we have a language $\mathcal{L}^*_{=,\mathrm{C}}$.

Symbols of $\mathcal{L}^*_{=,\mathrm{C}}$: All symbols of $\mathcal{L}^*_{\mathrm{C}}$ plus the symbol $=$.

Formulas of $\mathcal{L}^*_{=,\mathrm{C}}$: Modify the definition, given on page 24, of formulas of $\mathcal{L}^*_{\mathrm{C}}$ by renumbering clause (6) as clause (7) and adding the following clause.

(6) If t_1 and t_2 are variables or constants, then $t_1 = t_2$ is a formula.

Remark. Unique readability holds for $\mathcal{L}^*_{=,\mathrm{C}}$ by a proof very similar to the proof that it holds for $\mathcal{L}^*_{\mathrm{C}}$.

Models for $\mathcal{L}^*_{=,\mathrm{C}}$: Models for $\mathcal{L}^*_{=,\mathrm{C}}$ are the same as models for $\mathcal{L}^*_{\mathrm{C}}$.

Satisfaction and truth for $\mathcal{L}^*_{=,\mathrm{C}}$:

The notions of a *variable assignment* and of $\mathrm{den}^s_{\mathfrak{M}}$ are the same as for $\mathcal{L}^*_{\mathrm{C}}$. The definition of $v^s_{\mathfrak{M}}$ is the same as that for $\mathcal{L}^*_{=,\mathrm{C}}$, except that there is an extra subclause of the atomic clause (i):

$$\text{(c)}\ \ v^s_{\mathfrak{M}}(t_1 = t_2) = \begin{cases} \mathbf{T} & \text{if } \mathrm{den}^s_{\mathfrak{M}}(t_1) = \mathrm{den}^s_{\mathfrak{M}}(t_2); \\ \mathbf{F} & \text{if } \mathrm{den}^s_{\mathfrak{M}}(t_1) \neq \mathrm{den}^s_{\mathfrak{M}}(t_2). \end{cases}$$

Satisfaction and *truth* are defined as for $\mathcal{L}^*_{\mathrm{C}}$.

Logical implication for $\mathcal{L}^*_{=,\mathrm{C}}$: Logical implication, *validity*, and *satisfiability* are defined as for $\mathcal{L}^*_{\mathrm{C}}$.

Example. The following formulas are valid.

(a) $v_1 = v_1$	(d) $v_1 = v_2 \rightarrow (P^1 v_1 \leftrightarrow P^1 v_2)$
(b) $\forall v_1\, v_1 = v_1$	(e) $\forall v_1 \forall v_2 (v_1 = v_2 \rightarrow (P^1 v_1 \leftrightarrow P^1 v_2))$
(c) $\exists v_1\, v_1 = v_1$	(f) $v_1 = c \rightarrow (c = v_2 \rightarrow v_1 = v_2)$

The proof of the Compactness Theorem for $\mathcal{L}^*_{=,\mathrm{C}}$ is similar to that for $\mathcal{L}^*_{\mathrm{C}}$, but there is one important difference, as we shall see.

Lemma 4.1. *Let* Γ *be a finitely satisfiable set of sentences of* $\mathcal{L}^*_{=,\mathrm{C}}$ *and let* A *be a sentence of* $\mathcal{L}^*_{=,\mathrm{C}}$*. Then either* $\Gamma \cup \{A\}$ *is finitely satisfiable or* $\Gamma \cup \{\neg A\}$ *is finitely satisfiable.*

Proof. The proof is exactly like that of Lemma 3.6. $\square$

Lemma 4.2. *Let* Γ *be a finitely satisfiable set of sentences of* $\mathcal{L}^*_{=,\mathrm{C}}$*. Let* C^* *be a set gotten from* C *by adding infinitely many new constants. There is a set* Γ^* *of sentences of* $\mathcal{L}^*_{=,\mathrm{C}^*}$ *such that*

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

(2) Γ^* *is finitely satifiable ;*

(3) for every sentence A *of* $\mathcal{L}^*_{=,\mathrm{C}^*}$*, either* A *belongs to* Γ^* *or* $\neg A$ *belongs to* Γ^**;*

(4) Γ^* *is Henkin.*

Proof. The only change we have to make in the proof of Lemma 3.7 is that we must specify an alphabetical order for the symbols of $\mathcal{L}^*_{=,\mathrm{C}^*}$. Let us do this by letting the new symbol $=$ come immediately after $\forall$. $\square$

Lemma 4.3. *Let* Γ^* *be a set of sentences of a language* $\mathcal{L}^*_{=,\mathrm{C}^*}$ *having properties (2), (3), and (4) described in the statement of Lemma 4.2. Then* Γ^* *is satisfiable.*

Proof. We wish to begin, as we did in the proof of Lemma 3.8, by using Γ^* to define a model for $\mathcal{L}^*_{=,\mathrm{C}^*}$. But we must not define the model as we did before (on page 34). To see this, assume that we did use the old definition. Let c_1 and c_2 be two distinct members of C^* and suppose that the sentence $c_1 = c_2$ belongs to Γ^*, as is possible. Since $\chi(c_1) = c_1$, $\chi(c_2) = c_2$, and c_1 and c_2 are distinct objects, the new clause (i)(c) in the definition of $v^s_{\mathfrak{M}}$ implies that $v_{\mathfrak{M}}(c_1 = c_2) = \mathbf{F}$. Thus it is not the case that, for all formulas A, $v_{\mathfrak{M}}(A) = \mathbf{T}$ if and only if $A \in \Gamma^*$. But for the proof of Lemma 3.8 it was critical that this *was* the case. If we are to define a model for which it is the case, then we must make sure that

$$\text{if } c_1 = c_2 \in \Gamma^*, \text{ then } \chi(c_1) = \chi(c_2).$$

A 2-place relation R on a set X is an *equivalence relation on* X if and only if all three of the following conditions are satisfied.

(a) R is *reflexive*: Rxx holds for all $x \in X$.

(b) R is *symmetric*: if $x \in X$ and $y \in X$ and Rxy holds, then Ryx holds.

(c) R is *transitive*: if $x \in X$, $y \in X$, $z \in X$, and both Rxy and Ryz hold, then Rxz holds.

If R is an equivalence relation on X, then R divides X up into *equivalence classes*. For $x \in X$, let $[x]_R$, the *equivalence class of x with respect to R*, be defined by

$$[x]_R = \{y \in X \mid Rxy \text{ holds}\}.$$

Let R be the relation on C^* defined by

$$Rc_1c_2 \text{ holds} \;\text{ iff }\; c_1 = c_2 \in \Gamma^*.$$

We shall prove that R is an equivalence relation on C^*.

For reflexivity, we must show that $c = c$ belongs to Γ^* for all members c of C^*. Assume that $c = c \notin \Gamma^*$. By property (3) of Γ^*, $c \neq c \in \Gamma^*$, where we use $t \neq t'$ as an abbreviation for $\neg\, t = t'$. But then $\{c \neq c\}$ is a finite subset of Γ^* that is not satisfiable, contradicting (2).

For symmetry, we must show that, for all members c_1 and c_2 of Γ^*, if $c_1 = c_2 \in \Gamma^*$, then $c_2 = c_1 \in \Gamma^*$. Assume that $c_1 = c_2 \in \Gamma^*$ but $c_2 = c_1 \notin \Gamma^*$. Using (3), we get that $\{c_1 = c_2, c_2 \neq c_1\}$ is a finite subset of Γ^*. Once again, we contradict (2).

For transitivity, we must show that, for all members c_1, c_2, and c_3 of Γ^*, if $c_1 = c_2 \in \Gamma^*$ and $c_2 = c_3 \in \Gamma^*$, then $c_1 = c_3 \in \Gamma^*$. Assume that $c_1 = c_2 \in \Gamma^*$ and $c_2 = c_3 \in \Gamma^*$ but $c_1 = c_3 \notin \Gamma^*$. By (3), $\{c_1 = c_2, c_2 = c_3, c_1 \neq c_3\}$ is a finite subset of Γ^*, contradicting (2).

Define a model $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$ for $\mathcal{L}^*_{=,\mathrm{C}^*}$ as follows.

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

(ii) (a) $v(p_i) = \mathbf{T}$ if and only if $p_i \in \Gamma^*$.

(b) $v((P^n_i, [c_1]_R, \ldots, [c_n]_R)) = \mathbf{T}$ if and only if $P^n_i c_1 \ldots c_n \in \Gamma^*$.

(iii) $\chi(c) = [c]_R$ for each $c \in \mathrm{C}^*$.

To see that (ii)(b) is a genuine definition, we need to show that the truth-value it assigns does not depend on the choice of representatives c_j of the equivalence classes. To show this, assume that $[c_j]_R = [c'_j]_R$ for $1 \leq j \leq n$. By the definition of the equivalence classes, we have that $Rc_jc'_j$ holds for $1 \leq j \leq n$. By the definition of R, we get that the sentence $c_j = c'_j$ belongs to Γ^* for $1 \leq j \leq n$. We must show that $P^n_i c_1 \ldots c_n \in \Gamma^*$ if and only if

$P_i^n c'_1 \dots c'_n \in \Gamma^*$. For the "only if" direction, assume that $P_i^n c_1 \dots c_n \in \Gamma^*$ and that $P_i^n c'_1 \dots c'_n \notin \Gamma^*$. By (3), $\neg P_i^n c'_1 \dots c'_n \in \Gamma^*$. Thus

$$\{P_i^n c_1 \dots c_n,\, P_i^n c'_1 \dots c'_n,\, c_1 = c'_1, \dots, c_n = c'_n\}$$

is a finite subset of $\Gamma*$. By (2) it is satisfiable. This is a contradiction. The "if" direction is similar.

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

$$v_{\mathfrak{M}}(A) = \mathbf{T} \text{ if and only if } A \in \Gamma^* .$$

We prove, by the same variant of formula induction as we used in the proof of Lemma 3.8, that every sentence of $\mathcal{L}^*_{=,\mathrm{C}^*}$ has property P.

There are only two cases that are significantly different from the corresponding cases in the proof of Lemma 3.8, so we omit the other cases.

(i)(c) Atomic sentences $c_1 = c_2$ have P because

$$v_{\mathfrak{M}}(c_1 = c_2) = \mathbf{T} \text{ iff } \chi(c_1) = \chi(c_2) \text{ iff } [c_1]_R = [c_2]_R \text{ iff } c_1 = c_2 \in \Gamma^*.$$

(iv) Let A be a formula with no free occurrences of variables other than the variable x. Assume that, for every $c \in \mathrm{C}^*$, $A(x; c)$ has P. We prove that $\forall x A$ has P.

$$\begin{array}{lll}
v_{\mathfrak{M}}(\forall x A) = \mathbf{T} & \text{iff} & \text{for all } s,\ v^s_{\mathfrak{M}}(A) = \mathbf{T} \\
& \text{iff} & \text{for all } c \in \mathrm{C}^*, \text{ for all } s \text{ with } s(x) = [c]_R,\ v^s_{\mathfrak{M}}(A) = \mathbf{T} \\
& \text{iff} & \text{for all } c \in \mathrm{C}^*,\ v_{\mathfrak{M}}(A(x; c)) = \mathbf{T} \\
& \text{iff} & \text{for all } c \in \mathrm{C}^*,\ A(x; c) \in \Gamma^* \\
& \text{iff} & \forall x A \in \Gamma^*
\end{array}$$

The first "iff" is by the definition of $v_{\mathfrak{M}}$ and the fact that no variable besides x occurs free in A. The second "iff" is by the fact that no variable besides x occurs free in A and the fact that $\boldsymbol{D} = \{[c]_R \mid c \in \mathrm{C}^*\}$. The third "iff" is by the fact that $\chi(c) = [c]_R$ for each $c \in \mathrm{C}^*$. The fourth "iff" is by the fact that the sentences $A(x; c)$ have property P. The proof that the fifth "iff" holds is exactly the same as the corresponding step in the proof of Lemma 3.8.

Since, in particular, $v_{\mathfrak{M}}(A) = \mathbf{T}$ for every member of A of Γ^*, we have shown that Γ^* is satisfiable. □

The proof of the two Compactness Theorems that follow are just like the proofs Theorem 3.9 and Theorem 3.10.

Theorem 4.4 (Compactness). *Let Γ be a finitely satisfiable set of sentences of $\mathcal{L}^*_{=,\mathrm{C}}$. Then Γ is satisfiable, i.e., true in a model for $\mathcal{L}^*_{=,\mathrm{C}}$.*

Corollary 4.5 (Compactness, Second Form). *Let Γ be a set of sentences of $\mathcal{L}^*_{=,\mathrm{C}}$ and let A be a sentence such that $\Gamma \models A$. Then there is a finite subset Δ of Γ such that $\Delta \models A$.*

Exercise 4.1. Exhibit a sentence of $\mathcal{L}^*_{=,\emptyset}$ that is true in every model with exactly three elements and is false in all other models.

Exercise 4.2. Tell which of the following sentences of $\mathcal{L}^*_{=,\{\mathrm{c}\}}$ are valid. If a sentence is valid, explain briefly why. If it is invalid, give a model in which it is false.

(a) $\forall v_1\, c = c$
(b) $\forall v_1 \forall v_2 P^2 v_1 v_2 \rightarrow \forall v_1 \forall v_2\, v_1 = v_2$
(c) $\forall v_1 (P^1 v_1 \rightarrow \exists v_2 (v_1 = v_2 \wedge P^1 v_2))$
(d) $\forall v_1 (P^1 v_1 \rightarrow \forall v_2 (P^1 v_2 \rightarrow v_1 = v_2))$

The languages $\mathcal{L}^{\#}_{\mathrm{C}}$ of full first-order logic.

For each set C of constant symbols, we have a language $\mathcal{L}^{\#}_{\mathrm{C}}$.

Symbols of $\mathcal{L}^{\#}_{\mathrm{C}}$: All symbols of all symbols of $\mathcal{L}^*_{=,\mathrm{C}}$ plus *n-place function letters*

$$F^n_0,\ F^n_1,\ F^n_2, \ldots,$$

for each $n \geq 1$.

Terms of $\mathcal{L}^{\#}_{\mathrm{C}}$:

(1) Each variable or constant is a term.

(2) For each n and i, if $t_1, \ldots, t_n$ are terms, then $F^n_i t_1 \ldots t_n$ is a term.

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

Example of a term:

$$F^3_1 F^2_2 c F^1_0 v_4 v_6 F^1_0 c.$$

Remarks:

(a) As we shall see, terms are expressions that, in a model and under a variable assignment, denote a member of the domain of the model. The *terms* of $\mathcal{L}^*_{\mathrm{C}}$ and $\mathcal{L}^*_{=,\mathrm{C}}$ are—let us retroactively specify—the variables and constants. Variables and constants are the *atomic terms* of a language. The new ingredients of $\mathcal{L}^{\#}_{\mathrm{C}}$ are the complex terms given by clause (2) above.

(b) Just as we can do proof by formula induction and definition by recursion on formulas, so we can do *term induction* and *definition by recursion on terms*. In proving by term induction that all terms have a property P, we must (1) show that all variables and constants have P and (2) show that whenever $t_1, \ldots, t_n$ are terms that have P then $F_i^n t_1 \ldots t_n$ has P.

Formulas of $\mathcal{L}^{\#}_{\mathrm{C}}$*:* Replace clauses (2) and (6) in the definition of formulas of $\mathcal{L}^*_{=,\mathrm{C}}$ by the following clauses.

(2) For each n and i, if $t_1, \ldots, t_n$ are terms, then $P_i^n t_1 \ldots t_n$ is a formula.

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

Note that, with our retroactive definition of *term* for $\mathcal{L}^*_{\mathrm{C}}$ and $\mathcal{L}^*_{=,\mathrm{C}}$, the new clauses (2) and (6) have the same meaning as the old (2) and (6).

Remark. The proof of unique readability for formulas of $\mathcal{L}^{\#}_{\mathrm{C}}$ has a preliminary step. One first needs to prove *Unique Readability for Terms.* This states that every term is either a variable or constant or else is $F_i^n t_1 \ldots t_n$ for unique n, i, and $t_1, \ldots, t_n$. The rest of the proof of unique readability for formulas is similar to the proof for the other languages.

Models for $\mathcal{L}^{\#}_{\mathrm{C}}$*:*

A *model* for $\mathcal{L}^{\#}_{\mathrm{C}}$ is a triple $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$ satisfying conditions (i) and (ii) in the definition of a model for $\mathcal{L}^*_{=,C}$ and satisfying the following condition:

(iii) χ is a function that assigns

- (a) a member of $\boldsymbol{D}$ to each constant;
- (b) a member of $\boldsymbol{D}$ to each $(n+1)$-tuple of the form $(F_i^n, d_1, \ldots, d_n)$ for $d_1, \ldots, d_n$ members of $\boldsymbol{D}$.

Satisfaction, truth, and logical implication for $\mathcal{L}^{\#}_{\mathrm{C}}$*:*

The notion of a *variable assignment* is the same as for $\mathcal{L}^*_{\mathrm{C}}$ and $\mathcal{L}^*_{=,\mathrm{C}}$.

The definition of $\mathrm{den}^s_{\mathfrak{M}}$ for the other languages has to be extended so that $\mathrm{den}^s_{\mathfrak{M}}(t)$ is defined for all terms t. The definition is by recursion on terms.

(1) $\mathrm{den}^s_{\mathfrak{M}}(t) = s(t)$ if t is a variable, and $\mathrm{den}^s_{\mathfrak{M}}(t) = \chi(t)$ if t is a constant.

(2) $\mathrm{den}^s_{\mathfrak{M}}(F^n_i t_1 \ldots t_n) = \chi((F^n_i, \mathrm{den}^s_{\mathfrak{M}}(t_1), \ldots, \mathrm{den}^s_{\mathfrak{M}}(t_n)))$.

The definitions of *satisfaction*, *truth*, *logical implication*, *validity*, and *satisfiability* are word for word the same as for $\mathcal{L}^*_{=,\mathrm{C}}$.

The proof of the Compactness Theorem for $\mathcal{L}^{\#}_{\mathrm{C}}$ is very much like that for $\mathcal{L}^*_{=,\mathrm{C}}$. We list the lemmas and indicate the ways the proofs of the analogous earlier lemmas are to be modified.

Lemma 4.6. *Let Γ be a finitely satisfiable set of sentences of $\mathcal{L}^{\#}_{\mathrm{C}}$ and let A be a sentence of $\mathcal{L}^{\#}_{\mathrm{C}}$. Then either $\Gamma \cup \{A\}$ is finitely satisfiable or $\Gamma \cup \{\neg A\}$ is finitely satisfiable.*

Lemma 4.7. *Let Γ be a finitely satisfiable set of sentences of $\mathcal{L}^{\#}_{\mathrm{C}}$. Let C^* be a set gotten from C by adding infinitely many new constants. There is a set Γ^* of sentences of $\mathcal{L}^{\#}_{\mathrm{C}^*}$ such that*

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

(2) Γ^* *is finitely satifiable;*

(3) for every sentence A of $\mathcal{L}^{\#}_{\mathrm{C}^*}$, either A belongs to Γ^* or $\neg A$ belongs to Γ^*;

(4) Γ^* *is Henkin.*

Proof. The only change we have to make in the proof of Lemma 4.2 is that we must specify an alphabetical order for the symbols of $\mathcal{L}^{\#}_{\mathrm{C}^*}$. Let us do this by letting the new symbols F^n_i come after the symbols of $\mathcal{L}^*_{=,\mathrm{C}^*}$, ordered first by superscript and then by subscript. □

Lemma 4.8. *Let Γ^* be a set of sentences of a language $\mathcal{L}^{\#}_{\mathrm{C}^*}$ having properties (2), (3), and (4) described in the statement of Lemma 4.7. Then Γ^* is satisfiable.*

Proof. We need to make two additions to the proof of Lemma 4.3.

First we let

$$\chi((F^n_i, [c_1]_R, \ldots, [c_n]_R)) = [c]_R \quad \text{iff} \quad F^n_i c_1 \ldots c_n = c \in \Gamma^*.$$

It fairly easy to see that the definition does not depend on the choice of elements of equivalence classes. It is also easy to show—and we need to do so—that for all $c_1, \ldots, c_n$, there is a c such that $F^n_i c_1 \ldots c_n = c \in \Gamma^*$.

We also need to change clause (ii)(b) to make it analogous to the clause above.

The other change we have to make is in the atomic cases (i)(b) and (i)(c) of the proof that all formulas have property P.

Before considering the proofs of these facts, we first prove another fact that we will need in these proofs.

Say that a term t containing no variables has property Q if and only if, for every $c \in \mathbf{C}^*$,

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

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

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

(2) Assume that $t_1, \ldots, t_n$ have Q and let t be $F^n_i t_1 \ldots t_n$. Let $\mathrm{den}_{\mathfrak{M}}(t_i) = [c_i]_R$ for $1 \le i \le n$. Since the t_i have Q, the sentence $t_i = c_i$ belongs to Γ^* for each i. Let $\mathrm{den}_{\mathfrak{M}}(t) = [c]_R$. By the definition of $\mathrm{den}_{\mathfrak{M}}$, it follows that

$$\begin{aligned} \mathrm{den}_{\mathfrak{M}}(F^n_i c_1 \ldots c_n) &= \chi((F^n_i, [c_1]_R, \ldots, [c_n]_R)) \\ &= \mathrm{den}_{\mathfrak{M}}(F^n_i t_1 \ldots t_n) \\ &= \mathrm{den}_{\mathfrak{M}}(t) \\ &= [c]_R. \end{aligned}$$

By the definition of $v((F^n_i, [c_1]_R, \ldots, [c_n]_R))$, we have that $F^n_i c_1 \ldots c_n = c$ belongs to Γ^*. Assume that $F^n_i t_1 \ldots t_n = c$ does not belong to Γ^*. By property (3) of Γ^*, $F^n_i t_1 \ldots t_n \neq c$ belongs to Γ^*. Since $t_i = c_i \in \Gamma^*$ for every i, the set

$$\{t_1 = c_1, \ldots, t_n = c_n, F^n_i c_1 \ldots c_n = c, F^n_i t_1 \ldots t_n \neq c\}$$

is a finite subset of Γ^*. This set is not satisfiable, and so we have contradicted property (2) of Γ^*. In doing so, we have shown that t has Q.

Now we are ready for cases (i)(b) and (i)(c) of the property P proof. Let $\mathrm{den}_{\mathfrak{M}}(t_i) = [c_i]_R$ for $1 \le i \le n$. Since the t_i have Q, $t_i = c_i \in \Gamma^*$ for each i.

$$\begin{aligned} v_{\mathfrak{M}}(P^n_i t_1 \ldots t_n) = \mathbf{T} \quad &\text{iff} \quad v((P^n_i, \mathrm{den}_{\mathfrak{M}}(t_1), \mathrm{den}_{\mathfrak{M}}(t_n))) = \mathbf{T} \\ &\text{iff} \quad v((P^n_i, [c_1]_R, \ldots, [c_n]_R)) = \mathrm{T} \\ &\text{iff} \quad P^n_i c_1 \ldots c_n \in \Gamma^* \\ &\text{iff} \quad P^n_i t_1 \ldots t_n \in \Gamma^*, \end{aligned}$$

where the last iff is by properties (2) and (3) of Γ^*.

The proof for case (i)(c) is similar, and we omit it. $\square$

Theorem 4.9 (Compactness). *Let Γ be a finitely satisfiable set of sentences of $\mathcal{L}_{\mathrm{C}}^{\#}$. Then Γ is satisfiable, i.e., true in a model for $\mathcal{L}_{\mathrm{C}}^{\#}$.*

Corollary 4.10 (Compactness, Second Form). *Let Γ be a set of sentences of $\mathcal{L}_{\mathrm{C}}^{\#}$ and let A be a sentence such that $\Gamma \models A$. Then there is a finite subset Δ of Γ such that $\Delta \models A$.*

Exercise 4.3. Which of the following sentences are valid? For each one, explain or give (the relevant part of) a counter-model.

(a) $\exists v_1 F^3 v_2 c v_3 = v_1$

(b) $\forall v_1 \forall v_2 (v_1 \neq v_2 \rightarrow F^1 v_1 \neq F^1 v_2) \rightarrow \forall v_1 \exists v_2 F^1 v_2 = v_1$

Exercise 4.4. Give the omitted case (i)(c) in the proof of Lemma 4.8.

5 Deduction in First-Order Logic

The system $\mathbf{FOL}_{\mathrm{C}}$.

Let C be a set of constant symbols. $\mathbf{FOL}_{\mathrm{C}}$ is a system of deduction for the language $\mathcal{L}^{\#}_{\mathrm{C}}$.

Axioms: The following are axioms of $\mathbf{FOL}_{\mathrm{C}}$.

(1) All tautologies.

(2) Identity Axioms:

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

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

(3) Quantifier Axioms:

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

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

Rules of Inference:

Modus Ponens (MP) $\qquad \dfrac{A\,,\ (A \rightarrow B)}{B}$

Quantifier Rule (QR) $\qquad \dfrac{(A \rightarrow B)}{(A \rightarrow \forall x B)}$

provided the variable x does not occur free in A.

Discussion of the axioms and rules.

(1) We would have gotten an equivalent system of deduction if instead of taking all tautologies as axioms we had taken as axioms all instances (in $\mathcal{L}^{\#}_{\mathrm{C}}$) of the five schemas on page 13. All instances of these schemas are tautologies, so the change would have not have increased what we could deduce. In the

other direction, we can apply the proof of the Completeness Theorem for **SL** by thinking of all sententially atomic formulas as sentence letters. The proof so construed shows that every tautology in $\mathcal{L}_{\mathrm{C}}^{\#}$ is deducible using MP and schemas (1)–(5). Thus the change would not have decreased what we could deduce.

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

(3) 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 A 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.

(MP) Modus ponens is the rule we are familiar with from the system **SL**.

(QR) As we shall 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 shall 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 $\mathbf{FOL}_{\mathrm{C}}$ from a set Γ of sentences is a finite sequence $\mathbf{D}$ of formulas such that whenever a formula A occurs in the sequence $\mathbf{D}$ then at least one of the following holds.

(1) $A \in \Gamma$.
(2) A is an axiom.
(3) A 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* $\mathbf{FOL}_{\mathrm{C}}$ *of a formula* A *from a set* Γ of sentences is a deduction $\mathbf{D}$ in $\mathbf{FOL}_{\mathrm{C}}$ from Γ with A on the last line of $\mathbf{D}$. We write $\Gamma \vdash_{\mathbf{FOL}_{\mathrm{C}}} A$ and say A is *deducible* in $\mathbf{FOL}_{\mathrm{C}}$ from Γ to mean that there is a deduction in $\mathbf{FOL}_{\mathrm{C}}$ of A from Γ. We write $\vdash_{\mathbf{FOL}_{\mathrm{C}}} A$ for $\emptyset \vdash_{\mathbf{FOL}_{\mathrm{C}}} A$.

Announement. For the rest of this section, we shall omit subscripts "$\mathbf{FOL}_{\mathrm{C}}$." and phrases "in $\mathbf{FOL}_{\mathrm{C}}$" except in contexts where we are considering more than one set C.

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

Lemma 5.1. *Assume that $\Gamma \vdash A_i$ for $1 \leq i \leq n$ and $\{A_1, \ldots, A_n\} \models_{\mathrm{sl}} B$. Then $\Gamma \vdash B$. (See page 30 for the definition of $\models_{\mathrm{sl}}$.)*

Proof. If we string together deductions witnessing that $\Gamma \vdash A_i$ for each i, then we get a deduction from Γ in which each A_i is a line. The fact that $\{A_1, \ldots, A_n\} \models_{\mathrm{sl}} B$ gives us that the formula

$$(A_1 \to A_2 \to \cdots A_n \to B)$$

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

Lemma 5.1 justifies a derived rule, which we call SL. A formula B follows from formulas $A_1, \ldots, A_n$ by SL iff

$$\{A_1, \ldots, A_n\} \models_{\mathrm{sl}} B.$$

Lemma 5.2. *If $\Gamma \vdash A$ then $\Gamma \vdash \forall x A$.*

Proof. Assume that $\Gamma \vdash A$. Begin with a deduction from Γ with last line A. Use SL to get the line $(p_0 \vee \neg p_o) \to A$. Now apply QR to get $(p_0 \vee \neg p_o) \to \forall x A$. Finally use SL to get $\forall x A$. □

Lemma 5.2 justifies a derived rule, which we call Gen:

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

Lemma 5.3. *For all formulas A and B,*

$$\vdash \forall x(A \to B) \to (\forall x A \to \forall x B).$$

Proof. Here is an abbreviated deduction.

1.	$\forall x(A \to B) \to (A \to B)$	QAx
2.	$\forall x A \to A$	QAx
3.	$(\forall x(A \to B) \wedge \forall x A) \to B$	1,2; SL
4.	$(\forall x(A \to B) \wedge \forall x A) \to \forall x B$	3; QR
5.	$\forall x(A \to B) \to (\forall x A \to \forall x B)$	4; SL

□

Lemma 5.4. *For all formulas A,*

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

Proof. Here is an abbreviated deduction.

1.	$\forall y A \to A$	QAx
2.	$\neg A \to \neg \forall y A$	1; SL
3.	$\forall x(\neg A \to \neg \forall y A)$	2; Gen
4.	$\forall x(\neg A \to \neg \forall y A) \to (\forall x \neg A \to \forall x \neg \forall y A)$	Lemma 5.3
5.	$\forall x \neg A \to \forall x \neg \forall y A$	3,4; MP
6.	$\neg \forall x \neg \forall y A \to \neg \forall x \neg A$	5; SL
	$[\exists x \forall y A \to \exists x A]$	
7.	$\exists x \forall y A \to \forall y \exists x A$	6; QR

□

Exercise 5.1. Show that $\vdash (\exists v_1\, P^1 v_1 \to \exists v_2\, P^1 v_2)$.

Exercise 5.2. Show that $\{\forall v_1\, P^1 v_1\} \vdash \exists v_1\, P^1 v_1$.

Lemma 5.5. *If $\Gamma \vdash (A \to B)$ then $\Gamma \vdash (\forall\, x A \to \forall x\, B)$.*

Proof. Start with a deduction from Γ with last line $(A \to B)$. Use Gen to get the line $\forall x(A \to B)$. Then apply Lemma 5.3 and MP. □

Theorem 5.6 (Deduction Theorem). *Let Γ be a set of sentences, let A be a sentence, and let B be a formula. If $\Gamma \cup \{A\} \vdash B$ then $\Gamma \vdash (A \to B)$.*

Proof. The proof is similar to the proof of the Deduction Theorem for **SL**. Assume that $\Gamma \cup \{A\} \vdash B$. Let **D** be a deduction of B from $\Gamma \cup \{A\}$. We prove that

$$\Gamma \vdash (A \to C)$$

for every line C of **D**. Assume that this is false. Consider the first line C of **D** such that $\Gamma \nvdash (A \to C)$.

Assume that C either belongs to Γ or is an axiom. Then $\Gamma \vdash C$ and $(A \to C)$ follows from C by SL. Hence $\Gamma \vdash (A \to C)$.

Assume next that C is A. Since $A \to A$ is a tautology, $\Gamma \vdash (A \to A)$.

Assume next that C follows from formulas E and $(E \to C)$ by MP. These formulas are on earlier lines of **D** than C. Since C is the first "bad" line of **D**, $\Gamma \vdash A \to E$ and $\Gamma \vdash A \to (E \to C)$. Since

$$\{(A \to E), (A \to (E \to C))\} \models_{\rm sl} (A \to C),$$

$\Gamma \vdash (A \to C)$.

Finally assume that C is $(E \to \forall x F)$ and that C follows by QR from an earlier line $(E \to F)$ of $\mathbf{D}$. Since C is the first "bad" line of $\mathbf{D}$, $\Gamma \vdash A \to (E \to F)$. Starting with a deduction from Γ of $A \to (E \to F)$, we can get a deduction from Γ of $A \to (E \to \forall x F)$ as follows.

$\dots$	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$
n	$A \to (E \to F)$	$\dots$
$n+1.$	$(A \wedge E) \to F$	n; SL
$n+2.$	$(A \wedge E) \to \forall x F$	$n+1$; QR
$n+3.$	$A \to (E \to \forall x F)$	$n+2$; SL

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

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 $\Gamma \vdash (A \to B)$. $\square$

A set Γ of sentences of $\mathcal{L}^{\#}_{\mathrm{C}}$ is *inconsistent* in $\mathbf{FOL}_{\mathrm{C}}$ if there is a formula B such that $\Gamma \vdash_{\mathbf{FOL}_{\mathrm{C}}} B$ and $\Gamma \vdash_{\mathbf{FOL}_{\mathrm{C}}} \neg B$. Otherwise Γ is *consistent*.

Theorem 5.7. *Let Γ and Δ be sets of sentences, let A and $A_1, \dots, A_n$ be sentences, and let B be a formula.*

(1) $\Gamma \cup \{A\} \vdash B$ if and only if $\Gamma \vdash (A \to B)$.

(2) $\Gamma \cup \{A_1, \dots, A_n\} \vdash B$ if and only if $\Gamma \vdash (A_1 \to \dots \to A_n \to B)$.

(3) Γ is consistent if and only if there is some formula C such that $\Gamma \nvdash C$.

(4) If $\Gamma \vdash C$ for all $C \in \Delta$ and if $\Delta \vdash B$, then $\Gamma \vdash B$.

Proof. The proof is like the proof of Theorem 2.2, except that we may now use the derived rule SL instead of the particular axioms and rules of the system **SL**. $\square$

A system $\mathbf{S}$ of deduction for $\mathcal{L}^{\#}_{\mathrm{C}}$ is *sound* if, for all sets Γ of sentences and all formulas A, if $\Gamma \vdash_{\mathbf{S}} A$ then $\Gamma \models A$. A system $\mathbf{S}$ of deduction for $\mathcal{L}^{\#}_{\mathrm{C}}$ is *complete* if, for all sets Γ of sentences and all formulas A, if $\Gamma \models A$ then $\Gamma \vdash_{\mathbf{S}} A$.

Remark. These definitions are like the definitions of soundness and completeness of systems for $\mathcal{L}$, except that the new definitions require Γ to consist of sentences, not just formulas. We hereby make the analoguous definitions for our other languages.

Theorem 5.8 (Soundness). *The systems* $\mathbf{FOL}_{\mathrm{C}}$ *are sound.*

Proof. The proof is similar to the proof of soundness for **SL** (Theorem 2.4). Let $\mathbf{D}$ be a deduction in $\mathbf{FOL}_{\mathrm{C}}$ of a formula A from a set Γ of sentences. We shall show that, for every line C of $\mathbf{D}$, $\Gamma \models C$. Applying this to the last line of $\mathbf{D}$, this will give us that $\Gamma \models A$.

Assume that what we wish to show is false. Let C be the first line of $\mathbf{D}$ such that $\Gamma \not\models C$.

The cases that $C \in \Gamma$, that C is an axiom, and that C follows by MP from earlier lines of $\mathbf{D}$, are just like the corresponding cases in the proof of Theorem 2.4.

The only remaining case is that C is $B \to \forall x E$ and C follows by QR from an earlier line $B \to E$ of $\mathbf{D}$. Since C is the first "bad" line of $\mathbf{D}$, $\Gamma \models B \to E$. Let $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$ be any model and let s be any variable assignment. We assume that $v^s_{\mathfrak{M}}(\Gamma) = \mathbf{T}$ (i.e., that $v^s_{\mathfrak{M}}(H) = \mathbf{T}$ for each $H \in \Gamma$), and we show that $v^s_{\mathfrak{M}}(B \to \forall x E) = \mathbf{T}$. For this, we assume that $v^s_{\mathfrak{M}}(B) = \mathbf{T}$ and we show that $v^s_{\mathfrak{M}}(\forall x E) = \mathbf{T}$. Let d be any element of $\boldsymbol{D}$ and let s' be any variable assignment that agrees with s except that $s'(x) = d$. We must show that $v^{s'}_{\mathfrak{M}}(E) = \mathbf{T}$. Since Γ is a set of sentences, $v^{s'}_{\mathfrak{M}}(\Gamma) = \mathbf{T}$. Since the variable x does not occur free in B, $v^{s'}_{\mathfrak{M}}(B) = \mathbf{T}$. Since $\Gamma \models B \to E$, it follows that $v^{s'}_{\mathfrak{M}}(E) = \mathbf{T}$ □

Lemma 5.9. *Let* Γ *be a set of sentences of* $\mathcal{L}^{\#}_{\mathrm{C}}$ *consistent in* $\mathbf{FOL}_{\mathrm{C}}$ *and let* A *be a sentence of* $\mathcal{L}^{\#}_{\mathrm{C}}$. *Then either* $\Gamma \cup \{A\}$ *is consistent in* $\mathbf{FOL}_{\mathrm{C}}$ *or* $\Gamma \cup \{\neg A\}$ *is consistent in* $\mathbf{FOL}_{\mathrm{C}}$.

Proof. The proof is like that of Lemma 2.5. □

Lemma 5.10. *Let* Γ *be set of sentences of* $\mathcal{L}^{\#}_{\mathrm{C}}$ *consistent in* $\mathbf{FOL}_{\mathrm{C}}$. *Let* C^* *be a set gotten from* C *by adding infinitely many new constants. There is a set* Γ^* *of sentences of* $\mathcal{L}^{\#}_{\mathrm{C}^*}$ *such that*

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

(2) Γ^* *is consistent in* $\mathbf{FOL}_{\mathrm{C}^*}$ *;*

(3) for every sentence A *of* $\mathcal{L}^{\#}_{\mathrm{C}^*}$, *either* A *belongs to* Γ^* *or* $\neg A$ *belongs to* Γ^**;*

(4) Γ^* *is Henkin.*

Proof. Let $c_0, c_1, c_2, \ldots$ be all the constants of $\mathcal{L}^{\#}_{\mathrm{C}^*}$. Let

$$A_0,\ A_1,\ A_2,\ A_3,\ \ldots$$

be the list (defined in the proof of Lemma 4.7) of all the sentences of $\mathcal{L}^{\#}_{\mathrm{C}^*}$. As in that proof we define, by recursion on natural numbers, a function that associates with each natural number n a set Γ_n of formulas.

Let $\Gamma_0 = \Gamma$.

As in the proofs of Lemmas 3.7, 4.2, and 4.7, we shall make sure that, for each n, at most two sentences belong to Γ_{n+1} but not to Γ_n. As in the earlier proofs, it follows that for each n only finitely many of the new constants occur in sentences in Γ_n.

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

$$\Gamma'_n = \begin{cases} \Gamma_n \cup \{A_n\} & \text{if } \Gamma_n \cup \{A_n\} \text{ is consistent in } \mathbf{FOL}_{\mathrm{C}^*}; \\ \Gamma_n \cup \{\neg A_n\} & \text{otherwise.} \end{cases}$$

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

(a) $\neg A_n \in \Gamma'_n$.

(b) A_n is $\forall x_n B_n$ for some variable x_n and formula B_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 Γ'_n. Such an i must exist, since only finitely many of the infinitely many new constants occur in sentences in Γ'_n. Let

$$\Gamma_{n+1} = \Gamma'_n \cup \{\neg B_n(x_n; c_{i_n})\}.$$

Let $\Gamma^* = \bigcup_n \Gamma_n$.

Because $\Gamma = \Gamma_0 \subseteq \Gamma^*$, Γ^* has property (1).

We prove by mathematical induction that Γ_n is consistent for each n.

Γ_0 (i.e., Γ) is consistent in $\mathbf{FOL}_{\mathrm{C}}$ by hypothesis, but we must prove that it is consistent in $\mathbf{FOL}_{\mathrm{C}^*}$. Observe that any deduction $\mathbf{D}$ from Γ in $\mathbf{FOL}_{\mathrm{C}^*}$ of a formula of $\mathcal{L}^{\#}_{\mathrm{C}}$ can be turned into a deduction from Γ in $\mathbf{FOL}_{\mathrm{C}}$ 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 Γ is inconsistent in $\mathbf{FOL}_{\mathrm{C}}$ if it is inconsistent in $\mathbf{FOL}_{\mathrm{C}^*}$.

Assume that Γ_n is consistent in $\mathbf{FOL}_{\mathrm{C}^*}$. Lemma 5.9 implies that Γ'_n is consistent. If $\Gamma_{n+1} = \Gamma'_n$, then Γ_{n+1} is consistent. Assume then that $\Gamma_{n+1} =$

$\Gamma'_n \cup \{\neg B_n(x_n; c_{i_n})\}$ and, in order to derive a contradiction, assume that Γ_{n+1} is not consistent. By Theorem 5.7, every formula of $\mathcal{L}^{\#}_{\mathrm{C}}$ is deducible from Γ_{n+1} in $\mathbf{FOL}_{\mathrm{C}^*}$. Hence $\Gamma_{n+1} \vdash_{\mathbf{FOL}_{\mathrm{C}^*}} (p_0 \wedge \neg p_0)$. In other words,

$$\Gamma'_n \cup \{\neg B_n(x_n; c_{i_n})\} \vdash_{\mathbf{FOL}_{\mathrm{C}^*}} (p_0 \wedge \neg p_0).$$

By the Deduction Theorem,

$$\Gamma'_n \vdash_{\mathbf{FOL}_{\mathrm{C}^*}} \neg B_n(x_n; c_{i_n}) \to (p_0 \wedge \neg p_0).$$

Let $\mathbf{D}$ be a deduction from Γ'_n in $\mathbf{FOL}_{\mathrm{C}^*}$ with last line $\neg B_n(x_n; c_{i_n}) \to (p_0 \wedge \neg p_0)$. 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 Γ'_n or in $\neg B_n$, $\mathbf{D}'$ is a deduction from Γ'_n in $\mathbf{FOL}_{\mathrm{C}^*}$ with last line $\neg B_n(x_n; y) \to (p_0 \wedge \neg p_0)$. We can turn $\mathbf{D}'$ into a deduction from Γ'_n in $\mathbf{FOL}_{\mathrm{C}^*}$ with last line $\neg\forall x_n B_n \to (p_0 \wedge \neg p_0)$ as follows.

...	...	...
...	...	...
...	...	...
$n.$	$\neg B_n(x_n; y) \to (p_0 \wedge \neg p_0)$	...
$n+1.$	$\neg(p_0 \wedge \neg p_0) \to B_n(x_n; y)$	n; SL
$n+2.$	$\neg(p_0 \wedge \neg p_0) \to \forall y B_n(x_n; y)$	$n+1$; QR
$n+3.$	$\forall y B_n(x_n; y) \to B_n$	QAx
$n+4.$	$\neg(p_0 \wedge \neg p_0) \to B_n$	$n+2$,$n+3$; SL
$n+5.$	$\neg(p_0 \wedge \neg p_0) \to \forall x_n B_n$	$n+4$; QR
$n+6.$	$\neg\forall x_n B_n \to (p_0 \wedge \neg p_0)$	$n+5$; SL

This shows that $\Gamma'_n \vdash_{\mathbf{FOL}_{\mathrm{C}^*}} \neg\forall x_n B_n \to (p_0 \wedge \neg p_0)$. But $\Gamma'_n = \Gamma \cup \{\neg\forall x_n B_n\}$, so it follows that $\Gamma'_n \vdash_{\mathbf{FOL}_{\mathrm{C}^*}} (p_0 \wedge \neg p_0)$. By SL, we get the contradiction that Γ'_n is inconsistent in $\mathbf{FOL}_{\mathrm{C}^*}$.

As in the proof of Lemma 2.6, the consistency of all the Γ_n implies that consistency of Γ^*. Hence Γ^* has property (2).

Because either A_n or $\neg A_n$ belongs to Γ_{n+1} for each n and because each $\Gamma_{n+1} \subseteq \Gamma^*$, Γ^* has property (3).

If $A_n \notin \Gamma^*$, then $A_n \notin \Gamma_{n+1}$ and so $\neg A_n \in \Gamma_{n+1}$. But this implies that $\neg B_n(x_n; c_{i_n}) \in \Gamma_{n+1} \subseteq \Gamma^*$ if $A_n = \forall x_n B_n$. Hence Γ^* has property (4). □

Exercise 5.3. Show that

$$\{\forall v_1 \forall v_2 (P^2 v_1 v_2 \vee P^2 v_2 v_1)\} \vdash \forall v_1 P^2 v_1 v_1.$$

Exercise 5.4. Show that

$$\vdash \forall v_1 \exists v_2 F^1 v_1 = v_2.$$

Exercise 5.5. Let c_1 and c_2 be constants. Show that

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

Lemma 5.11. *Let Γ^* be a set of sentences of a language $\mathcal{L}^{\#}_{\mathrm{C}^*}$ having properties (2), (3), and (4) described in the statement of Lemma 5.10. Then Γ^* is satisfiable.*

Proof. We first note a useful fact about Γ^*.

(†) For all sentences A of $\mathcal{L}^{\#}_{\mathbf{C}^*}$, if $\Gamma^* \vdash A$ then $A \in \Gamma^*$.

To see why (†) holds, assume that $\Gamma^* \vdash A$ but $A \notin \Gamma^*$. By (3), $\neg A \in \Gamma^*$. Thus $\Gamma^* \vdash \neg A$, contradicting (2).

Remark. Though we did not state it, the analogue of (†) held for the set Γ^* of Lemma 2.7.

As in the proofs of Lemmas 4.3 and 4.8, we shall define a model whose domain is a set of equivalence classes of constants. As in the proof of Lemma 4.3, let R be the relation on C^* defined by

$$Rc_1c_2 \text{ holds iff } c_1 = c_2 \in \Gamma^*.$$

We shall prove that R is an equivalence relation on C^*.

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

For symmetry, we must show that, for all members c_1 and c_2 of Γ^*, if $c_1 = c_2 \in \Gamma^*$, then $c_2 = c_1 \in \Gamma^*$. Assume that $c_1 = c_2 \in \Gamma^*$. By Exercise 5.5, $\Gamma^* \vdash c_2 = c_1$. By (†), $c_2 = c_1 \in \Gamma^*$.

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 5.5
4.	$c_2 = c_1 \to (c_2 = c_3 \to c_1 = c_3)$	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 Γ^*, if $c_1 = c_2 \in \Gamma^*$ and $c_2 = c_3 \in \Gamma^*$, then $c_1 = c_3 \in \Gamma^*$. Assume that $c_1 = c_2 \in \Gamma^*$ and $c_2 = c_3 \in \Gamma^*$. By what we have just proved, $\Gamma^* \vdash c_1 = c_3$. By (†), $c_1 = c_3 \in \Gamma^*$.

We define a model $\mathfrak{M} = (\boldsymbol{D}, v, \chi)$ exactly as in the proof of Lemma 4.8, that is:

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

(ii) (a) $v(p_i) = \mathbf{T}$ if and only if $p_i \in \Gamma^*$.

(b) $v((P_i^n, [c_1]_R, \ldots, [c_n]_R)) = \mathbf{T}$ if and only if $P_i^n c_1 \ldots c_n \in \Gamma^*$.

(iii) (a) $\chi(c) = [c]_R$ for each $c \in \mathrm{C}^*$.

(b) $\chi((F_i^n, [c_1]_R, \ldots, [c_n]_R)) = [c]_R$ if and only if $F_i^n c_1 \ldots c_n = c \in \Gamma^*$.

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

A special case of the proof that clause (iii)(b) is independent of the choice of elements of equivalence classes is Exercise 5.6, and the proof for the general case is merely an elaboration of the proof for the special case. The case of (ii)(b) is a bit simpler.

The additional fact we to show concerning clause (iii)(b) is that, for all F_i^n and all $c_1, \ldots c_n$, that there is a c such that

$$F_i^n c_1 \ldots c_n = c \in \Gamma^*.$$

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

$$F_i^n c_1 \ldots c_n \neq c \in \Gamma^*.$$

By property (4) of Γ^*,

$$\forall v_1 F_i^n c_1 \ldots c_n \neq v_1 \in \Gamma^*.$$

Since

$$\forall v_1 F_i^n c_1 \ldots c_n \neq v_1 \in \Gamma^* \to F_i^n c_1 \ldots c_n \neq F_i^n c_1 \ldots c_n$$

is an instance of the Quantifier Axiom Schema,

$$\Gamma^* \vdash F_i^n c_1 \ldots c_n \neq F_i^n c_1 \ldots c_n.$$

But $F_i^n c_1 \ldots c_n \neq F_i^n c_1 \ldots c_n$ is an instance of Identity Axiom Schema (a), and so Γ^* is inconsistent, contradicting property (2) of Γ^*.

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

$$v_{\mathfrak{M}}(A) = \mathbf{T} \ \text{ if and only if } \ A \in \Gamma^*\,.$$

As in earlier proofs, we use a variant of formula induction to show that every sentence has property P.

The case of atomic sentences is like that case in the proof of Lemma 4.8, except for one change. Recall that in proving atomic cases (i)(b) and (i)(c), we first used a variant of term induction to demonstrate that all terms without variables have property Q, where t has property Q if and only if, for every $c \in \mathbf{C}^*$,

$$\text{if } \mathrm{den}_{\mathfrak{M}}(t) = [c]_R \text{ then } t = c \in \Gamma^*.$$

In the course of this demonstration, we got a contradiction from the assumption that $\Delta \subseteq \Gamma^*$, where

$$\Delta = \{t_1 = c_1, \ldots, t_n = c_n, F_i^n t_1 \ldots t_n = c, F_i^n c_1 \ldots c_n \neq c\}.$$

This assumption contradicted the hypothesis that Γ^* was finitely satisfiable. What we need to show in our new context is that it contradicts the hypothesis that Γ^* is consistent. Obviously $\Delta \vdash F_i^n c_1 \ldots c_n \neq c$. Thus it is enough to show that $\Delta \vdash F_i^n c_1 \ldots c_n = c$.

1.	$t_1 = c_1$	Premise
..	$\ldots$	$\ldots$
..	$\ldots$	$\ldots$
..	$\ldots$	$\ldots$
n.	$t_n = c_n$	Premise
$n+1$.	$t_1 = c_1 \rightarrow$	
	$(F_i^n t_1 t_2 \ldots t_{n-1} t_n = c \rightarrow F^n c_1 t_2 \ldots t_{n-1} t_n = c)$	IdAx(b)
..	$\ldots$	$\ldots$
..	$\ldots$	$\ldots$
..	$\ldots$	$\ldots$
$2n$.	$t_n = c_n \rightarrow$	
	$(F_i^n c_1 c_2 \ldots c_{n-1} t_n = c \rightarrow F^n c_1 c_2 \ldots c_{n-1} c_n = c)$	IdAx(b)
$2n+1$.	$F_i^n c_1 \ldots c_n = c$	1,. . . ,2n; SL

Cases cases (ii) and (iii) of the proof that all formulas have property P are like the corresponding cases in the proof of Lemma 2.7.

Case (iv) is like the corresponding case in the proof of Lemma 4.8, except for one change. The last step in case (iv) proof was to show that

$$\text{for all } c \in \mathrm{C}^*,\ A(x;c) \in \Gamma^* \quad \text{iff} \quad \forall x A \in \Gamma^*.$$

The "if" part of this "iff" was proved using the fact that Γ^* was finitely satisfiable. In the new context, we must prove it using the fact that Γ^* is consistent. To do this, assume that $\forall x A \in \Gamma^*$. Notice that, for each $c \in \mathrm{C}^*$, the sentence

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

is an instance of the Quantifier Axiom Schema. Thus $\Gamma^* \vdash A(x;c)$. By (†), $A(x;c) \in \Gamma^*$.

As in our earlier proofs, we have in particular that $v_{\mathfrak{M}}(A) = \mathbf{T}$ for every member of A of Γ^*, and this means we have shown that Γ^* is satisfiable. □

Theorem 5.12. *Let Γ be a consistent set of sentences of $\mathcal{L}_{\mathrm{C}}^{\#}$. Then Γ is satisfiable.*

Proof. By Lemma 5.10, let Γ^* have properties (1)–(3) of that lemma. By Lemma 2.7, Γ^* is satisfiable. Hence Γ is satisfiable. □

Theorem 5.13 (Completeness). *Let Γ be a set of sentences of $\mathcal{L}_{\mathrm{C}}^{\#}$ and let A be a formula of $\mathcal{L}_{\mathrm{C}}^{\#}$ such that $\Gamma \models A$. Then $\Gamma \vdash_{\mathbf{FOL}_{\mathrm{C}}} A$. In other words, $\mathbf{FOL}_{\mathrm{C}}$ is complete.*

Proof. Since $\Gamma \models A$, for every model $\mathfrak{M}$ and every variable assignment s, if Γ is true in $\mathfrak{M}$, then $v_{\mathfrak{M}}^{s}(A) = \mathbf{T}$. Let $x_1, \ldots, x_n$ be all the variables occurring free in A. Let $\mathfrak{M}$ be any model in which Γ is true. For every variable assignment s, $v_{\mathfrak{M}}^{s}(A) = \mathbf{T}$. This means that $\forall x_1 \ldots \forall x_n A$ is true in $\mathfrak{M}$. Thus

$$\Gamma \models \forall x_1 \ldots \forall x_n A.$$

Since $\Gamma \models \forall x_1 \ldots \forall x_n A$, $\Gamma \cup \{\neg \forall x_1 \ldots \forall x_n A\}$ is not satisfiable. By Theorem 5.12, $\Gamma \cup \{\neg \forall x_1 \ldots \forall x_n A\}$ is inconsistent. Let B be a formula such that $\Gamma \cup \{\neg \forall x_1 \ldots \forall x_n A\} \vdash B$ and $\Gamma \cup \{\neg \forall x_1 \ldots \forall x_n A\} \vdash \neg B$. By the Deduction Theorem, $\Gamma \vdash (\neg \forall x_1 \ldots \forall x_n A \to B)$ and $\Gamma \vdash \neg \forall x_1 \ldots \forall x_n A \to \neg B)$. By SL, $\Gamma \vdash \forall x_1 \ldots \forall x_n A$. Using the Quantifier Axiom Schema and MP n times, we get that $\Gamma \vdash A$. □

Exercise 5.6. In the proof of Lemma 5.11, clause (iii)(b) of the definition of the model $\mathfrak{M}$ says that

$$\chi((F_i^n, [c_1]_R, \ldots, [c_n]_R)) = [c]_R \quad \text{iff} \quad F_i^n c_1 \ldots c_n = c \in \Gamma^*.$$

Show, in the special case $n = 2$ and $i = 0$, that this definition does not depend on the choice of elements of equivalence classes. In other words, assume that

(1) $[c_1]_R = [c_1']_R$ and $[c_2]_R = [c_2']_R$;
(2) $F^2 c_1 c_2 = c \in \Gamma^*$ and $F^2 c_1' c_2' = c' \in \Gamma^*$,

and prove that

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

6 The semantics of second-order logic

The languages $\mathcal{L}^2_{\mathrm{C}}$ of second order logic.

For each set C of constant symbols, we have a language $\mathcal{L}^2_{\mathrm{C}}$.

Symbols of $\mathcal{L}^2_{\mathrm{C}}$: All symbols of $\mathcal{L}^{\#}_{\mathrm{C}}$ plus n-place *predicate variables*

$$V^n_0,\ V^n_1,\ V^n_2, \ldots$$

for each $n \geq 1$. In this section, we shall speak of $v_0, v_1, \ldots$ as *individual variables.*

Remark. It is common also to have n-place function variables, but we omit them in the interest of simplicity.

Terms of $\mathcal{L}^2_{\mathrm{C}}$: The definition of terms is the same as that for $\mathcal{L}^{\#}_{\mathrm{C}}$.

Formulas of $\mathcal{L}^2_{\mathrm{C}}$: Modify the definition of formulas of $\mathcal{L}^{\#}_{\mathrm{C}}$ by changing clauses (2) and (5) as follows.

(2) For each n and i, if $t_1, \ldots, t_n$ are terms, then $P^n_i t_1 \ldots t_n$ and $V^n_i t_1 \ldots t_n$ are formulas.

(5) If A is a formula and X is an individual or predicate variable, then $\forall X A$ is a formula.

Models for $\mathcal{L}^2_{\mathrm{C}}$: Models for $\mathcal{L}^2_{\mathrm{C}}$ are the same as models for $\mathcal{L}^{\#}_{\mathrm{C}}$.

Truth and logical implication for $\mathcal{L}^2_{\mathrm{C}}$:

For each model $\mathfrak{M} = (\mathbf{D}, v, \chi)$, a *variable assignment* is a function s that assigns an element of $\mathbf{D}$ to to each individual variable and an n-place relation on $\mathbf{D}$ to each n-place predicate variable. To the definition of $\mathrm{den}^s_{\mathfrak{M}}$, we add the stipulation that $\mathrm{den}^s_{\mathfrak{M}}(V^n_i) = s(V^n_i)$ for all n and i. The definition of $v^s_{\mathfrak{M}}$ is the same as that for $\mathcal{L}^{\#}_{\mathrm{C}}$, except for two changes. First, there is an extra subclause of the atomic clause (i):

(d) $v^s_{\mathfrak{M}}(V^n_i t_1 \ldots t_n) = \mathbf{T}$ if and only if $\mathrm{den}^s_{\mathfrak{M}}(V^n_i)(\mathrm{den}^s_{\mathfrak{M}}(t_1), \ldots, \mathrm{den}^s_{\mathfrak{M}}(t_n))$ holds.

Second, clause (iv) needs to be reinterpreted so that the variable x can be of either kind.

The definition of a *free occurrence* of a variable is as before, except that it now applies to both kinds of variables. *Satisfaction* and *truth* are defined as for $\mathcal{L}_{\mathrm{C}}^{\#}$, and so are *logical implication*, *validity*, and *satisfiability*.

Consider the following sentence of $\mathcal{L}_{\mathrm{C}}^{2}$.

$$\begin{aligned}\exists V^2(&\forall v_1 \exists v_2 V^2 v_1 v_2\\ &\wedge \forall v_1 \forall v_2 (V^2 v_1 v_2 \rightarrow \neg V^2 v_2 v_1)\\ &\wedge \forall v_1 \forall v_2 \forall v_3 ((V^2 v_1 v_2 \wedge V^2 v_2 v_3) \rightarrow V^2 v_1 v_3)).\end{aligned}$$

Call this sentence Inf. The solution to Exercise 3.3 shows that Inf can be true only in a model with an infinite domain. Conversely, Inf is true in every model with an infinite domain. This is because every infinite set can be linearly ordered in such a way that there is no greatest element. If the domain $\mathbf{D}$ of $\mathfrak{M}$ is infinite, then Inf is shown to be true in $\mathfrak{M}$ by the variable assignment that assigns such a linear ordering of $\mathbf{D}$ to V^2.

Theorem 6.1. *Compactness fails for $\mathcal{L}_{\mathrm{C}}^{2}$.*

Proof. For $n \geq 2$, let B_n be the following sentence of $\mathcal{L}_{\mathrm{C}}^{\#}$ (and so of $\mathcal{L}_{\mathrm{C}}^{2}$).

$$\exists v_1 \dots \exists v_n (v_1 \neq v_2 \wedge \dots \wedge v_1 \neq v_n \wedge v_2 \neq v_3 \wedge \dots \wedge v_{n-1} \neq v_n).$$

(There is a conjunct $v_i \neq v_j$ for all i and j such that $1 \leq i < j \leq n$.) For each n, B_n is true in all models whose domain has size $\geq n$ and it is false in all models whose domain has size $< n$. Let

$$\Gamma \quad = \quad \{\neg\mathrm{Inf}\} \cup \{B_2, B_3, B_4, \dots\}.$$

Clearly Γ is not satisfiable. The theorem will be proved if we can show that Γ is finitely satisfiable. Let Δ be a finite subset of Γ. Let n be the largest number such that $B_n \in \Delta$. If $\mathfrak{M}$ is any model whose domain is finite and has size $\geq n$, then, Δ is true in $\mathfrak{M}$. $\square$

Remark. Since compactness holds for $\mathcal{L}_{\mathrm{C}}^{\#}$, there can be no sentence like Inf in $\mathcal{L}_{\mathrm{C}}^{\#}$, and so also there is no sentence like $\neg$Inf in $\mathcal{L}_{\mathrm{C}}^{\#}$. Indeed, there is no set of sentences in $\mathcal{L}_{\mathrm{C}}^{\#}$ that does what $\neg$Inf does:

Exercise 6.1. Prove that there is no set Σ of sentence of $\mathcal{L}_{\mathrm{C}}^{\#}$ such that Σ is true in every model with a finite domain and false in every model with an infinite domain.

Hint. Consider the union of Σ and the set of all the B_n.

A *recursive* algorithm is an alogrithm that, except for limitations of program size and computer memory, could be implemented in a computer program and carried out by a computer. Call a formal language *decidable* if there is a recursive algorithm that, given a formula of the language as input, will output "yes" if the formula is valid and "no" if the formula is not valid.

The language of sentential logic is decidable. The truth-table alogrithm is recursive. The language of first-order logic is not decidable (even with empty C). Since the language of second-order logic contains that of first-order logic, the language of second-order logic cannot be decidable.

Let us call a formal language *semi-decidable* if there is a recursive algorithm that, given a formula of the language as input, will output "yes" if the formula is valid and will not say "yes" (and perhaps will not even halt) otherwise.

Decidability implies semi-decidability, so the language of sentential logic is semi-decidable. The language of first-order logic (with, say, only finitely many constants) is semi-decidable. It is not hard to see that there is a recursive alogrithm for listing all deductions from the empty set in the system $\mathbf{FOL}_{\mathrm{C}}$ (if C is finite). Given input A, run this listing algorithm and give output "yes" if a deduction with last line A is listed. The language of second-order logic is not semi-decidable, even with empty C empty.

Semi-decidable languages are essentially the same as the languages for which there exist usable sound and complete systems of deduction. Thus there is no such system of deduction for second-order logic.