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 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	$\neg, \rightarrow$
(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 $(A \to B)$.

(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. 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. An *open formula* is a formula with at least one free occurrence of a variable.

Example. The third occurrence of v_1 in the formula

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

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

The method of proof by induction on length applies to $\mathcal{L}^*_{\mathrm{C}}$ as it does to $\mathcal{L}$. The difference is that we now have four basic kinds of formulas.

Lemma 3.1. *For every formula A, 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 $(B \to C)$.

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

The proof of Lemma 3.1 is like that of Lemma 1.1. Because of the four kinds of formulas, there is an extra case in proofs by induction on length:

Case (iv) A is $\forall x B$ for some formula B and some variable x.

Also Case(i), the case that A is an atomic formula, has two subcases:

Case (i)(a) A is a sentence letter.

Case (i)(b) A is $P_i^n t_1 \dots t_n$ for some i and n, and some variables or constants $t, \dots, t_n$.

Unique readability holds for $\mathcal{L}_{\mathrm{C}}^*$ as it does for $\mathcal{L}$.

Lemma 3.2. *No proper initial segment of a formula of $\mathcal{L}_{\mathrm{C}}^*$is a formula of $\mathcal{L}_{\mathrm{C}}^*$.*

Theorem 3.3 (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 $(B \to C)$.

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

The proofs of Lemma 3.2 and Theorem 3.3 are similar to those of Lemma 1.2 and Theorem 1.3

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. And we might try to add a clause to v^* for formulas of the form $\forall x A$. There are complications with both of these ideas. We shall discuss the former complication now, the latter complication later.

The complication is that an open formula (one with free variables), does not, by itself, have a truth-value. Consider "$x < 3$," from the language of arithmetic, or "He is fat." from English. Neither formula has a truth-value on its own. To get a truth-value for the former, one needs to assign some particular number to the variable x. 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 open atomic formulas such as $P_1^2 v_3 c$. And, we cannot ignore the

semantics of formulas with free variables. For such formulas occur within the scope of $\forall$, as in $\forall v_2 P^1 v_2$, and the semantic properties of the quantified sentence depend upon the semantic properties of the formula. So in our semantics, we will allow for the assignment of a truth-value to $P^2_1 v_3 c$ given 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:

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

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

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

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.

Given a valuation v, in §1 we defined a function v^* that assigned truth-values to all the sentences in $\mathcal{L}$. It might seem natural to attempt to directly define a function $v_{\mathfrak{M}}$ that assigns a truth-value to all the formulas in $\mathcal{L}^*_{\mathrm{C}}$. In fact, we shall proceed in a more indirect fashion with our semantics for $\mathcal{L}^*_{\mathrm{C}}$. The first problem for this attempt arises with open formulas. As already noted, formulas such as $P^1 v_0$ or $\forall v_1 P^2 v_1 v_0$ do not have truth-values. They only have truth-values relative to assignments to their free variables. This problem might not seem fundamental. Could we directly define a function, just for all the *sentences* of $\mathcal{L}^*_{\mathrm{C}}$? We cannot even do that. Consider sentences of the form $\forall x A$. The natural clause is:

$$v_{\mathfrak{M}}(\forall x A) = \begin{cases} \mathbf{T} & \text{if for all } d \in \boldsymbol{D},\ v_{\mathfrak{M}}(A) = \mathbf{T} \\ \mathbf{F} & \text{otherwise}. \end{cases}$$

But, the contemplated recursive definition of $v_{\mathfrak{M}}$ does not give a value for a formula A relative to an assignment of d to x in A. So, we will need to define a function $v^s_{\mathfrak{M}}(A)$ that yields a truth-value for a formula A, given a model $\mathfrak{M}$, and an assignment s of values to the formula's free variables. *Then*, we can define $v_{\mathfrak{M}}$ for sentences, in terms of $v^s_{\mathfrak{M}}$.

So, 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. And, 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 the version of recursion on formulas appropriate for $\mathcal{L}^*_{\mathrm{C}}$, 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) $v^s_{\mathfrak{M}}(P^n_i t_1 \dots t_n) = v((P^n_i, \mathrm{den}^s_{\mathfrak{M}}(t_1), \dots, \mathrm{den}^s_{\mathfrak{M}}(t_n)))$;

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

(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 $v^s_{\mathfrak{M}}(A) = \mathbf{T}$.

Clause (iv) merits discussion, which it will receive shortly. 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:

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

Clause (iv) can be difficult to understand on a first encounter. What is the s' doing? What is the relation between s and s'? Clause (iv) is as it is to allow us to consider all assignments to x while holding fixed s's assignments to other variables. To understand this in more depth, we will start with an example in the language of mathematics, "For every number x, $x \geq y$". Suppose n is assigned to y. Given that assignment, the formula is true if

and only if each number in the domain is greater than or equal to n. That is, the formula is true iff for each number m, "$x \geq y$" is true if m is assigned to x and n is assigned to y. In this last statement, it becomes clear that we hold the assignment of n to y fixed, and consider different assignments to x. In determining the truth of "For every number x, $x \geq y$" given n as y's value, we ignore assignments of other values to y.

Now, consider an example from $\mathcal{L}^*_{\mathrm{C}}$, $\forall v_0 P^2 v_0 v_1$. Let $s(v_1) = e$, where $e \in \boldsymbol{D}$ of $\mathfrak{M}$. Intuitively, $\mathfrak{M}$ satisfies $\forall v_0 P^2 v_0 v_1$ under s if and only if for each s', if $s(v_1) = e$, then, whatever $s(v_0)$ may be, $v^{s'}_{\mathfrak{M}}(P^2 v_0 v_1) = \mathbf{T}$. That is, we consider any variable assignment s', and hence assignment of any values to v_0, so long as s' also assigns e to v_1. If $P^2 v_0 v_1$ is true in $\mathfrak{M}$ under all of those assignments, then $\forall v_0 P^2 v_0 v_1$ is true in $\mathfrak{M}$ under s. Otherwise, $\forall v_0 P^2 v_0 v_1$ is not.

These examples illustrate that our clause defining $v^s_{\mathfrak{M}}(\forall x A)$ needs to accomplish two things. It needs to hold fixed the assignments to free variable in A other than x. And it needs to consider all assignments to x. In clause (iv), we consider an assignment s', if it agree with s on assignments to all variables other than x. We thereby consider all s' that agree with all of s's assignments to all of A's free variables, other than x.

We now introduce some terminology. Note 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.[3] 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 $\mathfrak{M}$.

Example. Here are some sentences true in the model $\mathfrak{M}_a$ described on page 31: $\neg P^1 c$; $\forall v_1 \exists v_2\, P^2 v_1 v_2$; $\exists v_1 (p_2 \wedge P^2 v_1 v_1)$.

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) $\forall v_1 (P^1 v_1 \rightarrow p_2)$ $\quad$ (d) $\exists v_1 (P^1 v_1 \rightarrow p_2)$

[3] This is easily verifiable. See exercise 3.2.

Exercise 3.2. 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. For example, $\forall v_1 P^2 v_1 v_2(v_2; c) = \forall v_1 P^2 v_1 c$.

Let A be any formula such that x is the only variable with free occurrences in A. Show that, for every constant t, model $\mathfrak{M}$, and assignments s, s',

$$\text{if } \mathrm{den}^s_{\mathfrak{M}}(x) = \mathrm{den}^{s'}_{\mathfrak{M}}(t), \text{ then } v^s_{\mathfrak{M}}(A) = v^{s'}_{\mathfrak{M}}(A(x;t)).$$

Remarks:

1. Suppose that x is the only free variable in A, and that $s(x) = s'(x)$. Then, $\mathrm{den}^s_{\mathfrak{M}}(x) = \mathrm{den}^{s'}_{\mathfrak{M}}(x)$. $v^s_{\mathfrak{M}}(A) = v^{s'}_{\mathfrak{M}}(A(x;t))$. Since $A(x;x) = A$, $v^s_{\mathfrak{M}}(A) = v^{s'}_{\mathfrak{M}}(A)$. That is, whether or not $\mathfrak{M}$ satisfies A under s does not depend on the whole of s but only on the values $s(x)$.

2. Exercise 3.2 straightforwardly extends to cases in which multiple free variables $x_1, ..., x_n$ occur in A. A consequence of the generalization is that, for any formula A, whether or not $\mathfrak{M}$ satisfies A under s does not depend on the whole of s but only on the values $s(x_1), ..., s(x_n)$.

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 3.3. 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 3.4. 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) $\forall v_1 \forall v_2 (P^2 v_1 v_2 \rightarrow \neg P^2 v_2 v_1)$.

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

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

Exercise 3.5. 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. Consider:

(1) $\forall v_1 P^1 v_1 \rightarrow \forall v_1 P^1 v_1$;

(2) $\forall v_1 P^1 v_1 \rightarrow \exists v_1 P^1 v_1$.

Intuitively, (1) is valid just because of its truth-functional structure. If the antecedent is true, then the consequent is true, so the conditional is true. If the antecedent is false, the conditional is true. On the other hand, the validity of (2) depends also on its quantificational structure. So we would like to distinguish valid sentences like (1) from valid sentences like (2). We now explain how to do so.

Call a formula of any of our formal languages *sententially atomic* if it is neither a negation nor a conditional, i.e., if it is not $\neg A$ for any A and it is not $(A \rightarrow B)$ 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) $v^*((A \to B)) = \begin{cases} \mathbf{T} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{T}; \\ \mathbf{F} & \text{if } v^*(A) = \mathbf{T} \text{ and } v^*(B) = \mathbf{F}; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{T}; \\ \mathbf{T} & \text{if } v^*(A) = \mathbf{F} \text{ and } v^*(B) = \mathbf{F}. \end{cases}$

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

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 $\mathcal{L}^*_{\text{C}}$ every tautology is valid. The converse, while true for $\mathcal{L}$, is false for $\mathcal{L}^*_{\text{C}}$. For example, (2) is not a tautology but is valid.

Exercise 3.6.

$$\forall v_1(P^1 v_1 \to P^1 v_1)$$

is valid. Is it a tautology? Explain.

Exercise 3.7. 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: $(\forall v_2 P^1 v_2 \vee \exists v_1\, P^1 v_1)$.

Compactness

We now begin the proof of the Compactness Theorem for $\mathcal{L}^*_{\text{C}}$. As we did with $\mathcal{L}$, we call a set Γ of formulas of $\mathcal{L}^*_{\text{C}}$ *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 3.4. *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 1.4. 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. □

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 1.5. 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 3.6, the analogue of Lemma 1.6. A set Γ of formulas in a language $\mathcal{L}^*_{\mathrm{C}}$ is *Henkin in* $\mathcal{L}^*_{\mathrm{C}}$ 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 3.5. *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 $\mathcal{L}^*_{C^*}$, either A belongs to Γ^* or $\neg A$ belongs to Γ^*;

(4) Γ^* is Henkin in $\mathcal{L}^*_{C^*}$.

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

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

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

Now we give a method for listing all the sentences of $\mathcal{L}^*_{C^*}$ 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 $\mathcal{L}^*_{C^*}$, 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 $\mathcal{L}^*_{C^*}$. For each n, Γ_n will be a subset of Γ_{n+1}, and there will be at most 2 sentences that belong to $\Gamma_{n+1} \setminus \Gamma_n$.

Let $\Gamma_0 = \Gamma$.

Our definition of Γ_{n+1} is more complicated than it was in Lemma 1.5, because we need to insure that Γ^* is Henkin.

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. To see that such an i must exist, recall that there are infinitely many new constants in C^*. Γ_0 is Γ, which contains none of the new constants. For each $m < n$, at most two sentences were added in the passage from Γ_n to Γ_{n+1}. Thus $\Gamma'_n \setminus \Gamma_0$ is finite, so 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.[4] Assume that Γ_n is finitely satisfiable. Lemma 3.4 implies that Γ'_n is finitely satisfiable. If $\Gamma_{n+1} = \Gamma'_n$, then Γ_{n+1} is finitely satisfiable. Assume then that (a) and (b) hold and so $\Gamma_{n+1} = \Gamma'_n \cup \{\neg B_n(x_n; c_{i_n})\}$ 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

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

Since c_{i_n} does not occur in B_n,

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

[4] 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 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). □

Lemma 3.6. *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 3.5. 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) $v((P_i^n, c_1, \ldots, c_n)) = \mathbf{T}$ if and only if $P_i^n c_1 \ldots c_n \in \Gamma^*$.

(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 induction on length that every sentence of $\mathcal{L}^*_{\mathrm{C}^*}$ has property P.

Case (i)(a). A is a sentence letter p_i. A has P because $v_{\mathfrak{M}}(p_i) = v(p_i)$ and $v(p_i) = T$ if and only if $p_i \in \Gamma^*$.

Case (i)(b). A is $P_i^n c_1 \ldots c_n$. A has P by clause (ii)(b) above and by

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

Case (ii)—the case that A is $\neg B$ for some B—and Case (iii)—the case that A is $(B \to C)$ for some B and C—are just like the corresponding steps of the proof of Lemma 1.6.

Case (iv). A is $\forall x B$ for some formula B and some variable x. We prove that $\forall x B$ has P.

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

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

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

The "only if" part of the last biconditional 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 3.7 (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 3.5, let Γ^* have properties (1)–(4) of that lemma. By Lemma 3.6, 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 3.8 (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 1.8. □

Exercise 3.8. Call a set Γ of formulas in a language $\mathcal{L}^*_{\mathrm{C}}$ *Henkin* in* $\mathcal{L}^*_{\mathrm{C}}$ if and only if, for each formula A and each variable x, 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 $c \in \mathrm{C}$.

Let Γ^* be a set of sentences in a language $\mathcal{L}^*_{\mathrm{C}^*}$ having properties (2) and (3) described in the statement of Lemma 3.5. Show that Γ^* is Henkin in $\mathcal{L}^*_{\mathrm{C}^*}$ if and only if it is Henkin* in $\mathcal{L}^*_{\mathrm{C}^*}$.

Exercise 3.9. 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 $\mathcal{L}^*_{\mathrm{C}}$.

Exercise 3.10. Suppose we added $\exists$ as an official symbol of our languages $\mathcal{L}^*_{\mathrm{C}}$, extending the definition of $v^{\mathrm{s}}_{\mathfrak{M}}$ using clause (v) on page 32. The part of the proof of Lemma 3.6 by induction on length would have an extra case. Supply this extra case.