Introduction to Metalogic[1]

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	$\neg, \rightarrow$
(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 $\wedge$, $\vee$, 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 $\rightarrow$ and replace it by $\vee$. 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 $\neg A$.

[1]These notes, written by the instructor, were improved by modifications due to Luca Struble.

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

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

Lemma 1.1. *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 $(B \to C)$.

Proof. We first show that every formula has at least one of these three forms. 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).

To see that every formula has at most one of the three forms, note that those of form (1) begin (and end) with a sentence letter, those of form (2) begin with $\neg$, and those of form (3) begin with (. □

Let us officially regard formulas as sequences of symbols. Thus the formula $(p_1 \to \neg p_2)$ is officially a sequence of length 6. The formula $\neg p_3$ is a sequence of length 2, so $\neg p_3$ is *shorter than* $(p_1 \to \neg p_2)$.

We will often want to prove that all formulas have some property P. A method for proving this is *induction on length*. To prove by induction on length that every formula has property P, we must show the following:

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

Why does (†) imply that all formulas have property P? We show the contrapositive. Assume that some formula does not have P. Then there is a smallest number n such that some formula A of length n does not have P. Such an A is a witness that (†) is false.

In proving (†), one assumes that A is a formula and that every formula shorter than A has property P. One then proves that A has P. There are three cases one must consider.

Case (i) A is sentence letter.

Case (ii) A is $\neg B$ for some formula B.

Case (iii) A is $(B \to C)$ for some formulas B and C.

Remark: It is important to warn of a common misunderstanding. Compare the following two statments.

(ii) For every formula B, if every *formula* shorter than $\neg B$ has property P then $\neg B$ has P.

(ii′) For every *sentence letter* B, if every formula shorter than $\neg B$ has property P then $\neg B$ has P.

(ii), not (ii′), is part of induction on length. The B in question in (ii) is an *arbitrarily complex formula*. It may contain any number of instances of $\neg$ and $\rightarrow$. In doing Case (ii) of our inductive proof, *all* we assume about B is is that all formulas shorter than $\neg B$—i.e., all formulas no longer than B—have P. We do *not* we assume that B is a sentence letter. Similar remarks apply to Case (iii).

Lemma 1.2. *No proper initial segment of a formula is a formula.*

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

Case (i) is trivial, since sentence letters have no proper initial segments. (We don't count the empty sequence.)

Case (ii). Assume that B is a formula and that every formula shorter than $\neg B$ has P. Let D be a proper initial segment of $\neg B$. Since $\neg$ is not a formula,we may assume that D is $\neg B'$, where B' is a proper initial segment of B. Since B is shorter than $\neg B$, B has P, Thus B' is not a formula. It follows from this fact and Lemma 1.1 that $\neg B'$ is not a formula.

Case (iii). Assume that B and C are formulas and than every formula shorter than $(B \rightarrow C)$ has property P. Let D be a proper initial segment of $(B \rightarrow C)$. Assume, in order to derive a contradiction, that D is a formula. D begins with the symbol (, and therefore D must be $(B' \rightarrow C')$ for some formulas B' and C'. But both B and B' are shorter than $(B \rightarrow C)$, so both have property P. This implies that neither is a proper initial segment of the other. Hence B and B' must be the same formula. From this it follows that C' is a proper initial segment of C. But C is shorter than $(B \rightarrow C)$, and so C' cannot be a proper initial segment of C. This contradiction completes the proof of the lemma. □

Theorem 1.3 (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 $(B \to C)$.

Proof. If A does not begin with a left parenthesis, then Lemma 1.1 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 $(B \to C)$. Assume that there are formulas B' and C' such that B' is different from B and A is $(B' \to C')$. Then one of B and B' must be a proper initial segment of the other, contradicting Lemma 1.2. □

Unique Readability is a statement of grammatical univocality. Natural language contains syntactically ambiguous sentences, such as "Visiting in-laws can be tiresome". This sentence could mean that when one visits in-laws, that activity can be tiresome. Or, it could mean that certain people, in-laws who are visiting, can be tiresome. These two readings result from the syntactic ambiguity of "visiting in-laws". Unique Readability tells us that formulas of $\mathcal{L}$ are not grammatically ambiguous. Grammatical univocality is necessary for semantic univocality. For, as we shall see, a sentence's semantic properties, such as its truth-value, depend on its grammatical structure.

Exercise 1.1. Prove by induction on length that, for every formula A, the number of occurrences of sentence letters in A is one more than the number of occurrences of $\to$ 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 and truth conditions 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 $\mathcal{L}$.

Our method of interpreting the sentence letters is determined by our interest in $\mathcal{L}$. One reason to introduce a language is to use it in description or communication. If that were our reason for introducing $\mathcal{L}$, then we might assign propositions to each sentence letter. But we have a different reason for introducing $\mathcal{L}$. We are going to use $\mathcal{L}$ to study logical relations, logical relations between sentences of $\mathcal{L}$. We have to find a method of interpretation that is appropriate to the study of logical relations among sentences of $\mathcal{L}$. One semantical logical relation we are interested in is *implication*. Intuitively, an argument is valid if and only if it is a matter of the logical forms of the argument's premises and conclusion that if the premises are true then the conclusion is true. We can rephrase this as follows: an argument is valid if and only if, *whatever the constituent sentence letters mean*,[2] it is a matter of logic that if the premises are true then the conclusion is true. Thus our method of interpreting sentence letters is that it will have to allow us to consider all possible meanings of the sentence letters.

The sentential logic we want to study is a *truth-functional* logic: the truth-value of a complex formula depends only on the truth-values of the sentence letters that occur in it. Hence all that matters about the meaning of a sentence letter is whether—when given that meaning—the sentence letter is true or false. Thus our interpretation only needs to provide truth-values to the sentence-letters, nothing more.

We are going 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 define a *valuation* v for $\mathcal{L}$ to be a *function* that assigns to each sentence letter of $\mathcal{L}$ a *truth-value* **T** or **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 $v^*(A) = v(A)$;

[2] As long the meaning is the kind of meaning sentences have. Sentences do not have the same kind of meaning as singular terms, so our method of interpretation will not assign the meaning of a singular term to a sentence.

If the meaning of a sentence is a proposition, then the definiens is: whatever propositions are expressed by the constituent sentence-letters, it is a matter of logic that the conclusion is true if the premises are. And our method of interpretation *could* assign propositions to sentences. In fact, it will not.

(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 valuation v if $v^*(A) = \mathbf{T}$ and to be *false* under v if $v^*(A) = \mathbf{F}$.

One might worry that our definition of v^* is circular, because v^* figures in the definiens (the defining part of the definition). An example will make it intuitively clear that clauses (a)–(c) determine what truth-value v^* assigns to any given formula. Consider

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

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

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

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

For h to be a function, it is enough that h have exactly one value for each object in its domain. So, it is enough if our proposed definition determines, for each object in the domain, exactly one value for h. So it is enough if the definition does two things:

(1) The definition provides at least one value for each object in the domain.

(2) The definition provides no more than one value for each object in the domain.

As we will now explain, even though v^* figures in the *definiens*, our definition succeeds in doing both.

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 $h((A \to B))$ 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, a *recursive definition* of a function h has two parts, a base clause, and one or more successor clause. A base clause explicitly states h's value for some set of objects in its domain—in this case, the sentence letters. A successor clause specifies h's value for an object in terms of h's value on other objects—in this case, our successor clauses specify h's value on A in terms of h's value(s) for A's constituent.

One is not in general guaranteed of producing a successful recursive definition just by writing down a base clause and some successor clauses. A function on a domain assigns exactly one value to each object in the domain. To proceed rigorously, one must prove that one's definition determines a value for each object in the domain. And, one must prove that one's definition determines only one value for each object in the domain.

Remaining somewhat imprecise, let us sketch how to show that h has exactly one value for each member of the domain. Suppose (a)–(c) have been done. Thus, we suppose that h has at least one value for each member in its domain. Let P be the property a formula A has iff (a)–(c) determine a unique value $h(A)$. We show by induction on length that every formula A has P. Let A be a formula and assume that every formula shorter than A has P. We are supposing that we have defined a unique value $h(A)$ for the sentence letters, so A has property P in the case that A is a sentence letter. Now assume that A is $\neg B$ for some formula B. Since B is shorter than A, B has P—i.e., there is a unique $h(B)$. But that is enough to show that there is a unique $h(A)$ given by clause (b) of the definition. Finally assume that A is $(B \to C)$ for some formulas B and C. Since B and C are shorter than A, there are unique $h(B)$ and $h(C)$. By Unique Readabilty, there are no formulas B' and C' different from B and C such that A is $(B' \to C')$. Thus there is a unique $h(A)$ given by clause (c).

It will be convenient to introduce some abbreviations:

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

Bear in mind that $\vee$, $\wedge$, 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 $\wedge$, $\vee$, and $\leftrightarrow$ obey the following rules:

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

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

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

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) $((p_0 \to p_3) \to (\neg p_5 \to \neg p_4))$.

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* $\Gamma \therefore A$ 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 $\Gamma \therefore A$ 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. v satisfies A if and only if $v^*(A) = \mathbf{T}$. And v satisfies Γ if and only if, for all $A \in \Gamma$, $v^*(A) = \mathbf{T}$.

Remarks:

(a) We have a notion of truth-under-v; A is true under v if and only $v^*(A) = \mathbf{T}$. We do not have an absolute and unrelativized notion of truth. We do not have a notion of truth that allows us to ask, "Is A true?", only "Is A true under v?". This is because of our method of interpretation fo sentence letters. All we have said about sentence-letters is that a sentence letter has a truth-value given a valuation. Our method of interpretation gives no sense to the question "Is p_0 true?"

(b) For many Γ and A, $\Gamma \not\models A$, and $\Gamma \not\models \neg A$. For example, $\{(p_0 \rightarrow p_1)\} \not\models p_0$, and $\{(p_0 \rightarrow p_1)\} \not\models \neg p_0$. More generally, there are

many Γ and A for which there is a v under which Γ and A are true and there is a v' that satisfies Γ and $\neg A$.

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 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) $\Gamma \cup \{A_1, \ldots, A_n\} \models B$ if and only if $\Gamma \models (A_1 \to \cdots \to A_n \to B)$.
(3) Γ 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 $(A_1 \to (\cdots \to (A_n \to B) \cdots))$.

Exercise 1.8. There is an important relation between satisfiability and logical implication.

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

Prove that this relation obtains.

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 $h(x) = 2x$ (in terms of the successor function $+1$).

Exercise 1.9. 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.

The Compactness Theorem

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 1.4. *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 to derive a contradiction neither $\Gamma \cup \{A\}$ nor $\Gamma \cup \{\neg A\}$ is finitely satisfiable. This means that there is finite subeset $\bar{\Delta}$ of $\Gamma \cup \{A\}$ that is not satisfiable and there is a finite subset $\bar{\Delta}'$ of $\Gamma \cup \{\neg A\}$ that is not satisfiable. Because Γ is finitely satisfiable, neither $\bar{\Delta}$ nor $\bar{\Delta}'$ can be a subset of Γ. Thus $A \in \bar{\Delta}$ and $\neg A \in \bar{\Delta}'$. Let Δ be the set of members

of $\bar{\Delta}$ other than A, and let Δ' be the set of members of $\bar{\Delta}'$ other than $\neg A$. Then $\Delta \cup \{A\} = \bar{\Delta}$ and $\Delta' \cup \{\neg A\} = \bar{\Delta}'$, and so neither $\Delta \cup \{A\}$ nor $\Delta' \cup \{\neg A\}$ is satisfiable. Both Δ and Δ' are finite subsets of Γ, so $\Delta \cup \Delta'$ is a finite subset of Γ. Since Γ is finitely satisfiable, $\Delta \cup \Delta'$ 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. □

Lemma 1.5. *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

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

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 $\Gamma^* = \bigcup_n \Gamma_n$. Here $\bigcup_n \Gamma_n$ is the union of all the Γ_n. That is, $A \in \bigcup_n \Gamma_n$ if and only if there is some n such that $A \in \Gamma_n$.

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

Γ_0 is finitely satisfiable. By Lemma 1.4, 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 1.6. *Let Γ^* be a set of formulas having properties (2) and (3) described in the statement of Lemma 1.5. 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 induction on length that every formula has property P. Let A be a formula and assume that every formula shorter than A has P.

Case (i). A is a sentence letter. A has P by the definition of v.

Case (ii). A is $\neg B$ for some formula B. We want to show that $v^*(\neg B) = \mathbf{T}$ if and only if $\neg B \in \Gamma^*$. Consider the following biconditionals.

$$\begin{aligned} v^*(\neg B) = \mathbf{T} &\text{ iff } v^*(B) = \mathbf{F} \\ v^*(B) = \mathbf{F} &\text{ iff } B \notin \Gamma^* \\ B \notin \Gamma^* &\text{ iff } \neg B \in \Gamma^*. \end{aligned}$$

These biconditionals imply that $v^*(\neg B) = \mathbf{T}$ if and only if $\neg B \in \Gamma^*$.

The first biconditional is true by definition of v^*. The second biconditional is true because B is shorter than A and so has P. To finish Case (ii), we need only prove the third biconditional.

For the "if" direction, assume that $\neg B \in \Gamma^*$. Note that $\{B, \neg B\}$ is finite and not satisfiable. It follows by (2) that $\{B, \neg B\} \nsubseteq \Gamma^*$. So either $B \notin \Gamma^*$ or $\neg B \notin \Gamma^*$. Since $\neg B \in \Gamma^*$, $B \notin \Gamma^*$. Now for for the "only if" direction, assume that $B \notin \Gamma^*$. By (3) we have that $B \in \Gamma^*$ or $\neg B \in \Gamma^*$. It follows that $\neg B \in \Gamma^*$.

Case (iii). A is $(B \to C)$ for some formulas B and C. We want to show that $v^*((B \to C)) = \mathbf{T}$ if and only if $(B \to C) \in \Gamma^*$. Consider the following biconditionals.

$$\begin{aligned} v^*((B \to C)) = \mathbf{T} \quad &\text{iff} \quad \text{if } v^*(B) = \mathbf{T} \text{ then } v^*(C) = \mathbf{T} \\ \text{if } v^*(B) = \mathbf{T} \text{ then } v^*(C) = \mathbf{T} \quad &\text{iff} \quad \text{if } B \in \Gamma^* \text{ then } C \in \Gamma^* \\ \text{if } B \in \Gamma^* \text{ then } C \in \Gamma^* \quad &\text{iff} \quad (B \to C) \in \Gamma^* . \end{aligned}$$

These biconditionals imply that

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

The first biconditional is true by definition of v^*. The second biconditional is true because B and C are shorter than $(B \to C)$ and sofollows from the first by the inductive assumption.

For the "if" direction, assume that $(B \to C) \in \Gamma^*$ and $B \in \Gamma^*$. Note that $\{(B \to C), B, \neg C\}$ is finite and not satisfiable. It follows by (2) that $\{(B \to C), B, \neg C\} \nsubseteq \Gamma^*$. Since $(B \to C) \in \Gamma^*$ and $B \in \Gamma^*$ by our assumption, $\neg C \notin \Gamma^*$. By (3), $C \in \Gamma^*$.

Now for the "only if" direction, assume that if $B \in \Gamma^*$ then $C \in \Gamma^*$. Either $B \in \Gamma^*$ or $B \notin \Gamma^*$. Assume first that $B \notin \Gamma^*$. By (3), $\neg B \in \Gamma^*$. Note that $\{\neg B, \neg(B \to C)\}$ is finite and not satisfiable. By (2) and (3), it follows that $(B \to C) \in \Gamma^*$. Next assume that $B \in \Gamma^*$ By our assumption, $C \in \Gamma^*$. Note that $(\neg(B \to C), C\}$ is not satisfiable. By (2) and (3), $(B \to C) \in \Gamma^*$.

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

Exercise 1.10. Suppose that we added $\wedge$ as an official symbol of $\mathcal{L}$, extending the definition of truth using the table for $\wedge$ on page 8. The

proof by induction on length would have an extra case case: the case that A is $B \wedge C$ for some formulas B and C. Supply this case for the proof by formula induction just given.

Theorem 1.7 (Compactness). *Let Γ be a finitely satisfiable set of formulas. Then Γ is satisfiable.*

Proof. By Lemma 1.5, let Γ^* have properties (1)–(3) of that lemma. By Lemma 1.6, Γ^* is satisfiable. Hence Γ is satisfiable. $\square$

Corollary 1.8 (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 1.11. Prove Corollary 1.8.