Logic for Philosophy

Theodore Sider

December 4, 2007
Preface

This book is an elementary introduction to the logic that students of contemporary philosophy ought to know. It covers i) basic approaches to logic, including proof theory and especially model theory, ii) extensions of standard logic (such as modal logic) that are important in philosophy, and iii) some elementary philosophy of logic. It prepares students to read the logically sophisticated articles in today’s philosophy journals, and helps them resist bullying by symbol-mongerers. In short, it teaches the logic necessary for being a contemporary philosopher.

For better or for worse (I think better), the last century-or-so’s developments in logic are part of the shared knowledge base of philosophers, and inform, in varying degrees of directness, every area of philosophy. Logic is part of our shared language and inheritance. The standard philosophy curriculum therefore includes a healthy dose of logic. This is a good thing. But the advanced logic that is part of this curriculum is usually a course in “mathematical logic”, which usually means an intensive course in metalogic (for example, a course based on the excellent Boolos and Jeffrey (1989).) I do believe in the value of such a course. But if advanced undergraduate philosophy majors or beginning graduate students are to have one advanced logic course, that course should not, I think, be a course in metalogic. The standard metalogic course is too mathematically demanding for the average philosophy student, and omits material that the average student needs to know. If there is to be only one advanced logic course, let it be a course designed to instill logical literacy.

I begin with a sketch of standard propositional and predicate logic (developed more formally than in a typical intro course.) I briefly discuss a few extensions and variations on each (e.g., three-valued logic, definite descriptions). I then discuss modal logic and counterfactual conditionals in detail. I presuppose familiarity with the contents of a typical intro logic course: the meanings of the logical symbols of first-order predicate logic without identity
or function symbols; truth tables; translations from English into propositional and predicate logic; some proof system (e.g., natural deduction) in propositional and predicate logic.

I drew heavily from the following sources, which would be good for supplemental reading:

- Propositional logic: Mendelson (1987)
- Descriptions, multi-valued logic: Gamut (1991a)
- Sequents: Lemmon (1965)
- Further quantifiers: Glanzberg (2006); Sher (1991, chapter 2); Westerståhl (1989); Boolos and Jeffrey (1989, chapter 18)
- Modal logic: Gamut (1991b); Cresswell and Hughes (1996)
- Semantics for intuitionism: Priest (2001)
- Counterfactuals: Lewis (1973)
- Two-dimensional modal logic: Davies and Humberstone (1980)

Another source was Ed Gettier’s 1988 modal logic class at the University of Massachusetts.

I am also deeply grateful for feedback from colleagues, and from students in courses on this material. In particular, Marcello Antosh, Josh Armstrong, Gabe Greenberg, Angela Harper, Sami Laine, Gregory Lavers, Alex Morgan, Jeff Russell, Brock Sides, Jason Turner, Crystal Tychonievich, Jennifer Wang, Brian Weatherson, and Evan Williams: thank you.
## Contents

**Preface**

1 **Nature of Logic**
   1.1 Logical consequence and logical truth .................................. 2
   1.2 Form and abstraction ......................................................... 2
   1.3 Formal logic .......................................................................... 6
   1.4 Correctness and application ................................................... 7
   1.5 The nature of logical consequence ......................................... 8
   1.6 Extensions, deviations, variations ......................................... 10
      1.6.1 Extensions ................................................................. 10
      1.6.2 Deviations ................................................................. 11
      1.6.3 Variations ................................................................. 11
   1.7 Metalogic, metalanguages, and formalization ............................ 12
   1.8 Set theory ........................................................................... 14

2 **Propositional Logic**
   2.1 Grammar of PL .................................................................... 18
   2.2 The semantic approach to logic ............................................. 21
   2.3 Semantics of PL .................................................................... 22
   2.4 Natural deduction in PL ........................................................ 28
      2.4.1 Sequents ................................................................. 29
      2.4.2 Rules ................................................................... 31
      2.4.3 Sequent proofs ......................................................... 34
      2.4.4 Example sequent proofs ............................................. 35
   2.5 Axiomatic proofs in PL ......................................................... 39
      2.5.1 Example axiomatic proofs .......................................... 42
      2.5.2 The deduction theorem ............................................... 47
   2.6 Soundness and completeness of PL ........................................ 47
CONTENTS

3 Variations and Deviations from PL 52
  3.1 Alternate connectives 52
    3.1.1 Symbolizing truth functions in propositional logic 52
    3.1.2 Inadequate connective sets 56
    3.1.3 Sheffer stroke 57
  3.2 Polish notation 57
  3.3 Multi-valued logic 58
    3.3.1 Łukasiewicz’s system 60
    3.3.2 Kleene’s “strong” tables 62
    3.3.3 Kleene’s “weak” tables (Bochvar’s tables) 63
    3.3.4 Supervaluationism 63
  3.4 Intuitionism 67

4 Predicate Logic 71
  4.1 Grammar of predicate logic 71
  4.2 Semantics of predicate logic 72
  4.3 Establishing validity and invalidity 77

5 Extensions of Predicate Logic 80
  5.1 Identity 80
    5.1.1 Grammar for the identity sign 80
    5.1.2 Semantics for the identity sign 81
    5.1.3 Symbolizations with the identity sign 81
  5.2 Function symbols 83
    5.2.1 Grammar for function symbols 85
    5.2.2 Semantics for function symbols 86
    5.2.3 Symbolizations with function symbols: some examples 88
  5.3 Definite descriptions 89
    5.3.1 Grammar for 89
    5.3.2 Semantics for 90
    5.3.3 Eliminability of function symbols and definite descriptions 92
  5.4 Further quantifiers 95
    5.4.1 Generalized monadic quantifiers 96
    5.4.2 Generalized binary quantifiers 98
    5.4.3 Second-order logic 100
CONTENTS

9.4 Countermodels and validity proofs in SQML . . . . . . . . . . . . 228
9.5 Philosophical questions about SQML . . . . . . . . . . . . . . . . 234
  9.5.1 The necessity of identity . . . . . . . . . . . . . . . . . . . . . . 234
  9.5.2 The necessity of existence . . . . . . . . . . . . . . . . . . . . . 236
  9.5.3 Necessary existence defended . . . . . . . . . . . . . . . . . . 241
9.6 Variable domains . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 244
  9.6.1 Countermodels to the Barcan and related formulas . . . . 246
  9.6.2 Expanding, shrinking domains . . . . . . . . . . . . . . . . . . . 248
  9.6.3 Strong and weak necessity . . . . . . . . . . . . . . . . . . . . . 249

10 Two-dimensional modal logic . . . . . . . . . . . . . . . . . . . . . . . . 253
  10.1 Actuality . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 253
    10.1.1 Kripke models with actual worlds . . . . . . . . . . . . . . . 254
    10.1.2 Semantics for @ . . . . . . . . . . . . . . . . . . . . . . . . . . . . 255
    10.1.3 Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 256
  10.2 × . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 257
    10.2.1 Two-dimensional semantics for × . . . . . . . . . . . . . . . 258
    10.2.2 Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 260
  10.3 Fixedly . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 262
    10.3.1 Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 262
  10.4 A philosophical application: necessity and a priority . . . . . 265

References . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 272
Chapter 1

Nature of Logic

Since you are reading this book, you are probably already familiar with some logic. You probably know how to translate English sentences into symbolic notation—into propositional logic:

<table>
<thead>
<tr>
<th>English</th>
<th>Propositional logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>If snow is white then grass is green</td>
<td>$S \rightarrow G$</td>
</tr>
<tr>
<td>Either snow is white or grass is not green</td>
<td>$S \lor \neg G$</td>
</tr>
</tbody>
</table>

and into predicate logic:

<table>
<thead>
<tr>
<th>English</th>
<th>Predicate logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Jones is happy then someone is happy</td>
<td>$H_j \rightarrow \exists x H_x$</td>
</tr>
<tr>
<td>Anyone who is friends with Jones is either</td>
<td>$\forall x [F x j \rightarrow (I x \lor \forall y F x y)]$</td>
</tr>
<tr>
<td>insane or friends with everyone</td>
<td></td>
</tr>
</tbody>
</table>

You are probably also familiar with some basic techniques for evaluating arguments written out in symbolic notation. You have probably encountered truth tables, and some form of proof theory (perhaps a “natural deduction” system; perhaps “truth trees”.) You may have even encountered some elementary model theory. In short: you had an introductory course in symbolic logic.

What you already have is: literacy in elementary logic. What you will get out of this book is: literacy in the rest of logic that philosophers tend to presuppose, plus a deeper grasp of what logic is all about.

So what is logic all about?
1.1 Logical consequence and logical truth

Logic is about logical consequence. The statement “someone is happy” is a logical consequence of the statement “Ted is happy”. If Ted is happy, then it logically follows that someone is happy. Put another way: the statement “Ted is happy” logically implies the statement “someone is happy”. Likewise, the statement “Ted is happy” is a logical consequence of the statements “It’s not the case that John is happy” and “Either John is happy or Ted is happy”. The first statement follows from the latter two statements. If the latter two statements are true, the former must be true. Put another way: the argument whose premises are the latter two statements, and whose conclusion is the former statement, is a logically correct one.\(^1\)

Relatedly, logic is about logical truth. A logical truth is a sentence that is “true purely by virtue of logic”. Examples might include: “it’s not the case that snow is white and also not white”, “All fish are fish”, and “If Ted is happy then someone is happy”. It is plausible that logical truth and logical consequence are related thus: a logical truth is a sentence that is a logical consequence of any sentences whatsoever.

1.2 Form and abstraction

Logicians focus on form. Consider again the following argument:

\[
\text{Argument A:} \quad \begin{align*}
\neg \phi & \quad \text{It’s not the case that John is happy} \\
\phi \lor \psi & \quad \text{Ted is happy or John is happy} \\
\therefore \psi & \quad \text{Therefore, Ted is happy}
\end{align*}
\]

Argument A is logically correct—its conclusion is a logical consequence of its premises. It is customary to say that this is so\(^1\) in virtue of its form—in virtue of the fact that its form is:

\[
\begin{align*}
\neg \phi & \\
\phi \lor \psi & \\
\therefore \psi
\end{align*}
\]

\(^1\)The word ‘valid’ is sometimes used for logically correct arguments, but I will reserve that word for a different concept: that of a logical truth according to the semantic conception of logical truth.
Likewise, we say that “it’s not the case that snow is white and snow is not white” is a logical truth because it has the form: it’s not the case that \( \phi \) and not-\( \phi \).

We need to think hard about the idea of form. Apparently, we got the alleged form of Argument A by replacing some words with Greek letters and leaving other words as they were. We replaced the sentences ‘John is happy’ and ‘Ted is happy’ with \( \phi \) and \( \psi \), respectively, but left the expressions ‘It’s not the case that’ and ‘or’ as they were, resulting in the schematic form displayed above. Let’s call that form, “Form 1”. What’s so special about Form 1? Couldn’t we make other choices for what to leave and what to replace? For instance, if we replace the predicate ‘is happy’ with the schematic letter \( \alpha \), leaving the rest intact, we get this:

Form 2:  
It’s not the case that John is \( \alpha \)  
Ted is \( \alpha \) or John is \( \alpha \)  
Therefore, Ted is \( \alpha \)

And if we replace the ‘or’ with the schematic letter \( \gamma \) and leave the rest intact, then we get this:

Form 3:  
It’s not the case that John is happy  
Ted is happy \( \gamma \) John is happy  
Therefore, Ted is happy

If we think of Argument A as having Form 1, then we can think of it as being logically correct in virtue of its form, since every “instance” of Form 1 is logically correct. That is, no matter what sentences we substitute in for the greek letters \( \phi \) and \( \psi \) in Form 1, the result is a logically correct argument. Now, if we think of Argument A as being Form 2, we can continue to think of Argument A as being logically correct in virtue of its form, since, like Form 1, every instance of Form 2 is logically correct: no matter what predicate we change \( \alpha \) to, Form 2 becomes a logically correct argument. But if we think of Argument A as being Form 3, then we cannot think of it as being logically correct in virtue of its form, for not every instance of Form 3 is a logically correct argument. If we change \( \gamma \) to ‘if and only if’, for example, then we get the following logically incorrect argument:

It’s not the case that John is happy  
Ted is happy if and only if John is happy  
Therefore, Ted is happy
So, what did we mean, when we said that Argument A is logically correct in virtue of its form? What is Argument A’s form? Is it Form 1, Form 2, or Form 3?

There is no such thing as the form of an argument. When we assign an argument a form, what we are doing is focusing on certain words and ignoring others. We leave intact the words we’re focusing on, and we insert schematic letters for the rest. Thus, in assigning Argument A Form 1, we’re focusing on the words (phrases) ‘it is not the case that’ and ‘or’, and ignoring other words. More generally, in (standard) propositional logic, we focus on the phrases ‘if…then’, ‘if and only if’, ‘and’, ‘or’, and so on, and ignore others. We do this in order to investigate the relations of logical consequence that hold in virtue of these words’ meaning. The fact that Argument A is logically correct depends just on the meaning of the phrases ‘it is not the case that’ and ‘or’; it does not depend on the meanings of the sentences ‘John is happy’ and ‘Ted is happy’. We can substitute any sentences we like for ‘φ’ and ‘ψ’ in Form 1 and still get a valid argument.

In predicate logic, on the other hand, we focus on further words: ‘all’ and ‘some’. Broadening our focus in this way allows us to capture a wider range of logical consequences and logical truths. For example “If Ted is happy then someone is happy” is a logical truth in virtue of the meaning of ‘someone’, but not merely in virtue of the meanings of the characteristic words of propositional logic.

Call the words on which we’re focusing—that is, the words that we leave intact when we construct the forms of sentences and arguments—the logical constants. (We can speak of natural language logical constants—‘and’, ‘or’, etc. for propositional logic; ‘all’ and ‘some’ in addition for predicate logic—as well as symbolic logical constants: ∧, ∨, etc. for propositional logic; ∀ and ∃ in addition for predicate logic.) What we’ve seen is that the forms we assign depend on what we’re considering to be the logical constants.

We call these expressions logical constants because we interpret them in a constant way in logic, in contrast to other terms. For example, ∧ is a logical constant; in propositional logic, it always stands for conjunction. There are fixed rules governing ∧, in proof systems (the rule that from \(P \land Q\) one can infer \(P\), for example), in the rules for constructing truth tables, and so on. Moreover, these rules are distinctive for ∧: there are different rules for other logical constants such as ∨. In contrast, the terms in logic that are not logical constants do not have fixed, particular rules governing their meanings. For example, there are no special rules governing what one can do with a \(P\) as
opposed to a $Q$ in proofs or truth tables. That’s because $P$ doesn’t symbolize any sentence in particular; it can stand for any old sentence.

There isn’t anything sacred about the choices of logical constants we make in propositional and predicate logic; and therefore, there isn’t anything sacred about the customary forms we assign to sentences. We could treat other words as logical constants. We could, for example, stop taking ‘or’ as a logical constant, and instead take ‘It’s not the case that John is happy’, ‘Ted is happy’, and ‘John is happy’ as logical constants. We would thereby view Argument A as having Form 3. This would not be a particularly productive choice (since it would not help to explain the correctness of Argument A), but it’s not wrong simply by virtue of the concept of form.

More interestingly, consider the fact that every argument of the following form is logically correct:

\begin{itemize}
  \item $\alpha$ is a bachelor
  \item Therefore, $\alpha$ is unmarried
\end{itemize}

Accordingly, we could treat the predicates ‘is a bachelor’ and ‘is unmarried’ as logical constants, and develop a corresponding logic. We could introduce special symbolic logical constants for these predicates, we could introduce distinctive rules governing these predicates in proofs. (The rule of “bachelor-elimination”, for instance, might allow one to infer “$\alpha$ is unmarried” from “$\alpha$ is a bachelor”.) As with the choices of the previous paragraph, this choice of what to treat as a logical constant is also not ruled out by the concept of form. And it would be more productive than the choices of the last paragraph. Still, it would be far less productive than the usual choices of logical constants in predicate and propositional logic. The word ‘bachelor’ doesn’t have as general application as the words commonly treated as logical constants in propositional and predicate logic; the latter are ubiquitous.

At least, this remark about “generality” is one idea about what should be considered a “logical constant”, and hence one idea about the scope of what is usually thought of as “logic”. Where to draw the boundaries of logic—and indeed, whether the logic/nonlogic boundary is an important one to draw—is an open philosophical question about logic. At any rate, in this course, one thing we’ll do is study systems that expand the list of logical constants from standard propositional and predicate logic.
1.3 Formal logic

Modern logic is “mathematical” or “formal” logic. This means simply that one studies logic using mathematical techniques. More carefully: in order to develop theories of logical consequence, and logical truth, one develops a formal language (see below), one treats the sentences of the formal language as mathematical objects; one uses the tools of mathematics (especially, the tools of very abstract mathematics, such as set theory) to formulate theories about the sentences in the formal language; and one applies mathematical standards of rigor to these theories. Mathematical logic was originally developed to study mathematical reasoning\(^2\), but its techniques are now applied to reasoning of all kinds.

Think, for example, of propositional logic (this will be our first topic below). The standard approach to analyzing the logical behavior of ‘and’, ‘or’, and so on, is to develop a certain formal language, the language of propositional logic. The sentences of this language look like this:

\[
P (Q\to R) \lor (Q\to \sim S) \\
P \leftrightarrow (P \land Q)
\]

The symbols \(\land\), \(\lor\), etc., are used to represent the English words ‘and’, ‘or’, and so on (the logical constants for propositional logic), and the sentence letters \(P, Q, R, \ldots\) are used to represent declarative English sentences.

Why ‘formal’? Because we stipulate, in a mathematically rigorous way, a grammar for the language; that is, we stipulate a mathematically rigorous definition of the idea of a sentence of this language. Moreover, since we are only interested in the logical behavior of the chosen logical constants ‘and’, ‘or’, and so on, we choose special symbols \((\land, \lor \ldots)\) for these words only; we use \(P, Q, R, \ldots\) indifferently to represent any English sentence whose internal logical structure we are willing to ignore.\(^3\)

We go on, then, to study (as always, in a mathematically rigorous way) various concepts that apply to the sentences in formal languages. In propositional

---

\(^2\)Notes

\(^3\)Natural languages like English also have a grammar, and the grammar can be studied using mathematical techniques. But the grammar is much more complicated, and is discovered rather than stipulated; and natural languages lack abstractions like the sentence letters.
logic, for example, one constructs a mathematically rigorous definition of a tautology (“all Trues in the truth table”), and a rigorous definition of a provable formula (e.g., in terms of a system of deduction, using rules of inference, assumptions, and so on).

Of course, the real goal is to apply the notions of logical consequence and logical truth to sentences of English and other natural languages. The formal languages are merely a tool; we need to apply the tool.

1.4 Correctness and application

To apply the tools we develop for formal languages, we need to speak of a formal system as being correct. What does that sort of claim mean?

As we saw, logicians use formal languages and formal structures to study logical consequence and logical truth. And the range of structures that one could in principle study is very wide. For example, I could introduce a new notion of “provability” by saying “in Ted Logic, the following rule may be used when constructing proofs: if you have \( P \) on a line, you may infer \( \sim P \). The annotation is ‘T’.” I could then go on to investigate the properties of such a system. Logic can be viewed as a branch of mathematics, and we can mathematically study any system we like, including a system (like Ted logic) in which one can “prove” \( \sim P \) from \( P \).

But no such formal system would shed light on genuine logical consequence and genuine logical truth. It would be implausible to claim, for example, that when we translate an English argument into symbols, the conclusion of the resulting symbolic argument may be derived in Ted logic from its premises iff the conclusion of the original English argument is a logical consequence of its premises.

Thus, the existence of a coherent, specifiable logical system must be distinguished from its application. When we say that a logical system is correct, we have in mind some application of that system. Here’s an oversimplified account of one such correctness claim. Suppose we have developed a certain formal system for constructing proofs in propositional logic. And suppose we have specified some translation scheme from English into the language of propositional logic. This translation schema would translate English ‘and’ into the logical \( \land \), English ‘or’ into the logical \( \lor \), and so on. Then, the claim that the formal system gives a correct logic of English ‘and’, ‘or’, etc. might be taken to be the claim that one English sentence is a logical consequence of
some other English sentences in virtue of ‘and’, ‘or’, etc., iff one can prove the translation of the former English sentence from the translations of the latter English sentences in the formal system.

In this book I won’t spend much time on philosophical questions about which formal systems are correct. My goal is rather to introduce those formalisms that are ubiquitous in philosophy, to give you the tools you need to address such philosophical questions yourself. Still, from time to time, we’ll dip just a bit into these philosophical questions, in order to motivate our choices of logical systems to study.

1.5 The nature of logical consequence

The previous section discussed what it means to say that a formal theory gives a correct account of logical consequence (as applied to sentences of English and other natural languages). But what is it for sentences to stand in the relation of logical consequence? What is logical consequence?

The question here is a philosophical question, as opposed to a mathematical one. Logicians define various notions concerning sentences of formal languages: derivability in this or that proof-system, “all trues in the truth table”, and so on. They thereby stipulatively introduce various formal concepts. These formal concepts are good insofar as they correctly model logical truth and logical consequence. But in what do logical truth and logical consequence—the intuitive concepts, as opposed to the stipulatively introduced concepts—consist? This is one of the core questions of philosophical logic.

This book is not primarily a book in philosophical logic, so we won’t spend much time on the question. However, I do want to make clear that the question is indeed a question. The question is sometimes obscured by the fact that terms like ‘logical truth’ are often stipulatively defined in logic books. This can lead to the belief that there are no genuine issues concerning these notions. It is also obscured by the fact that one philosophical theory of these notions—the model-theoretic one—is so dominant that one can forget that it is a nontrivial theory. Stipulative definitions are of course not things whose truth can be questioned; but stipulative definitions of logical notions are good insofar as the stipulated notions accurately model the real, intuitive, nonstipulated notions of logical consequence and logical truth. Further, the stipulated definitions generally concern formal languages, whereas the ultimate goal is an understanding of correct reasoning of the sort that we actually do, using natural languages.
Let’s focus just on logical consequence. Here is a quick survey of some competing philosophical accounts of its nature. Probably the most standard account is the semantic, or model-theoretic one. Intuitively, a logical truth is “true no matter what”. The model theoretic account is one way of making this slogan precise. It says that $\phi$ is a logical consequence of the sentences in set $\Gamma$ if the formal translation of $\phi$ is true in every model (interpretation) in which the formal translations of the members of $\Gamma$ are true. This account needs to be spelled out in various ways. First, “formal translations” are translations into a formal language; but which formal language? It will be a language that has a logical constant for each English logical expression. But that raises the question of which expressions of English are logical expressions. In addition to ‘and’, ‘or’, ‘all’, and so on, are any of the following logical expressions?

- necessarily
- it will be the case that
- most
- it is morally wrong that

Further, the notion of translation must be defined; further, an appropriate definition of ‘model’ must be chosen.

Similar issues of refinement confront a second account, the proof-theoretic account: $\phi$ is a logical consequence of the members of $\Gamma$ iff the translation of $\phi$ is provable from the translations of the members of $\Gamma$. We must decide what formal language to translate into, and we must decide upon an appropriate account of provability.

A third view is Quine’s: $\phi$ is a logical consequence of the members of $\Gamma$ iff there is no way to (uniformly) substitute new nonlogical expressions for nonlogical expressions in $\phi$ and the members of $\Gamma$ so that the members of $\Gamma$ become true and $\phi$ becomes false.

Three other accounts should be mentioned. The first account is a modal one. Say that $\Gamma$ modally implies $\phi$ iff it is not possible for $\phi$ to be false while the members of $\Gamma$ are true. (What does ‘possible’ mean here? There are many kinds of possibility one might have in mind: so-called “metaphysical possibility”, “absolute possibility”, “idealized epistemic possibility”…. Clearly the acceptability of the proposal depends on the legitimacy of these notions. We discuss modality later in the book, beginning in chapter 6.) One might then propose that $\phi$ is a logical consequence of the members of $\Gamma$ iff $\Gamma$ modally implies $\phi$. (An
intermediate proposal: \( \phi \) is a logical consequence of the members of \( \Gamma \) iff, in virtue of the forms of \( \phi \) and the members of \( \Gamma \), \( \Gamma \) modally implies \( \phi \). More carefully: \( \phi \) is a logical consequence of the members of \( \Gamma \) iff \( \Gamma \) modally implies \( \phi \), and moreover, whenever \( \Gamma' \) and \( \phi' \) result from \( \Gamma \) and \( \phi \) by (uniform) substitution of nonlogical expressions, \( \Gamma' \) modally implies \( \phi' \). This is like Quine’s definition, but with modal implication in place of truth-preservation.) Second, there is a primitivist account, according to which logical consequence is a primitive notion. Third, there is a pluralist account according to which there is no one kind of genuine logical consequence. There are, of course, the various concepts proposed by each account, each of which is trying to capture genuine logical consequence; but in fact there is no further notion of genuine logical consequence at all; there are only the proposed construals.

As I say, this is not a book on philosophical logic, and so we will not inquire further into which (if any) of these accounts is correct. We will, rather, focus exclusively on two kinds of formal proposals for modeling logical consequence and logical truth: model-theoretic and proof-theoretic proposals.

1.6 Extensions, deviations, variations

“Standard logic” is what is usually studied in introductory logic courses. It includes propositional logic (logical constants: \( \land, \lor, \neg, \rightarrow, \leftrightarrow \)), and predicate logic (logical constants: \( \forall, \exists, \) variables). In this book we’ll consider various modifications of standard logic:

1.6.1 Extensions

Here we add to standard logic. We add both:

- new symbols
- new cases of logical consequence and logical truth that we can model

We do this in order to get a better representation of logical consequence. There is more to logic than that captured by plain old standard logic.

We extended propositional logic, after all, to get predicate logic. You can do a lot with propositional logic, but you can’t capture the obvious fact that

---

4See Gamut (1991a, pp. 156-158).
‘Ted is happy’ logically implies ‘someone is happy’ using propositional logic alone. It was for this reason that we added quantifiers, variables, predicates, etc., to propositional logic (added symbols), and added means to deal with these new symbols in semantics and proof theory (new cases of logical consequence and logical truth). But there is no need to stop with plain old predicate logic. We will consider adding symbols for identity, function symbols, and definite descriptions to predicate logic, for example, and we’ll add a sign for “necessarily” when we get to modal logic. And in each case, we’ll introduce modifications to our formal theories that let us account for logical truths and logical consequences involving the new symbols.

1.6.2 Deviations

Here we change, rather than add. We retain the same symbols from standard logic, but we alter standard logic’s proof theory and semantics. We therefore change what we say about the logical consequences and logical truths that involve the symbols.

Why do this? Perhaps because we think that standard logicians are wrong about what the right logic for English is. If we want to correctly model logical consequence in English, therefore, we must construct systems that behave differently from standard logic.

For example, in the standard semantics for propositional logic, every formula is either true or false. But some have argued that natural language sentences like the following are neither true nor false:

- The king of the United States is bald
- Sherlock Holmes weighs more than 178 pounds
- Bill Clinton is tall
- There will be a sea battle tomorrow

If so, perhaps we should abandon the standard semantics for propositional logic in favor of multi-valued logic, in which formulas are allowed to be neither true nor false.

1.6.3 Variations

Here we also change standard logic, but we change the notation without changing the content of logic. We study alternate ways of expressing the same thing.
For example, in intro logic we show how:

\[ \neg (P \land Q) \]
\[ \neg P \lor \neg Q \]

are two different ways of saying the same thing. We will study other ways of saying what those two sentences say, including:

\[ P|Q \]
\[ \neg \land PQ \]

In the first case, \( | \) is a new symbol for “not both”. In the second case (“Polish notation”), the \( \neg \) and the \( \land \) mean what they mean in standard logic; but instead of going between the \( P \) and the \( Q \), the \( \land \) goes before \( P \) and \( Q \). The value of this, as we’ll see, is that we no longer will need parentheses.

### 1.7 Metalogic, metalanguages, and formalization

In introductory logic, we learned how to *use* certain logical systems. We learned how to do truth tables, construct derivations, and so on. But logicians do not spend much of their time developing systems only to sit around all day doing derivations in those systems. As soon as a logician develops a new system, he or she will begin to ask questions about that system. For an analogy, imagine people who make up games. They might invent a new version of chess. Now, they might spend some time actually playing the new game. But if they were like logicians, they would soon get bored with this and start asking questions about the game, such as: “is the average length of this new game longer than the average length of a game of standard chess?” “Is there any strategy one could pursue which will guarantee a victory?” Analogously, logicians ask questions like: what things can be proved in such and such a system? Can you prove the same things in this system as in system \( X \)? Proving things about logical systems is part of “meta-logic”, which is an important part of logic.

One particularly important question of metalogic is that of *soundness* and *completeness*. Standard textbooks introduce a pair of methods for characterizing logical truth for the formulas of propositional logic. One is semantic: a formula is a semantic logical truth iff the truth table for that formula has all “trues” in its final column. Another is proof-theoretic: a sentence is a proof-theoretic logical truth iff there exists a derivation of it (from no premises), where a derivation
is then appropriately defined. (Think: introduction- and elimination- rules, conditional and indirect proof, and so on.) The question of soundness and completeness is: how do these two methods for characterizing logical truth relate to each other? The question is answered, in the case of propositional logic, by the following metalogical results, which are proved in standard books on metalogic:

**Soundness of propositional logic:** Any proof-theoretic logical truth is a semantic logical truth

**Completeness of propositional logic:** Any semantic logical truth is a proof-theoretic logical truth

These are really interesting claims! They show that the method of truth tables and the method of constructing derivations amount to the same thing, as applied to symbolic formulas of propositional logic. One can establish similar results for standard predicate logic.

A couple remarks about proving things in metalogic.

*First:* what do we mean by “proving”? We do *not* mean: constructing a derivation in the logical system we’re investigating. We’re trying to construct a proof about the system. We do this in *English*, and we do it with informal (though rigorous!) reasoning of the sort one would encounter in a mathematics book. Logicians often distinguish the “object language” from the “metalanguage”. The object language is the language that’s being studied—the language of propositional logic, for example. Sentences of this object language look like this:

\[
P \land Q
\]

\[
\sim(P \lor Q) \leftrightarrow R
\]

The metalanguage is the language we use to talk about the object language. In the case of the present book, the metalanguage is English. Here are some example sentences of the metalanguage:

‘*P \land Q*’ is a sentence with three symbols, one of which is a logical constant

Every sentence of propositional logic has the same number of left parentheses as right parentheses
If there exists a derivation of a formula, then its truth table contains all “trues” in its final column (i.e., soundness)

Thus, we formulate metalogical claims in the metalanguage, and our proofs in metalogic take place in the metalanguage.

Second: to get anywhere in metalogic, we will have to get picky about a few things about which one can afford to be lax in introductory logic. Let’s look at soundness, for instance. To be able to prove this, in a mathematically rigorous way, we’ll need to have the terms in it defined very carefully. In particular, we’ll need to say exactly what we mean by ‘sentence of propositional logic’, ‘truth tables’, and ‘derived’. Defining these terms precisely (another thing we’ll do using English, the metalanguage!) is known as formalizing logic. Our first task will be to formalize propositional logic.

1.8 Set theory

As mentioned above, modern logic uses mathematical techniques to study formal languages. The mathematical techniques in question are those of “set theory”. Only the most elementary set-theoretic concepts and assumptions will be needed, and you are probably already familiar with them; but nevertheless, here is a brief overview.

Sets have members. Consider, for example, the set, \( \mathbb{N} \), of natural numbers. Each natural number is a member of \( \mathbb{N} \): 1 is a member of \( \mathbb{N} \), 2 is a member of \( \mathbb{N} \), and so on. We use the expression “\( \in \)” for this relationship of membership; thus, we can say: \( 1 \in \mathbb{N} \), \( 2 \in \mathbb{N} \), and so on. We often name a set by putting names of its members between braces: \( \{1, 2, 3, 4,...\} \) is another name of \( \mathbb{N} \).

Sets are not limited to sets of mathematical entities; anything can be a member of a set. Thus, we may speak of the set of people, the set of cities, or—to draw nearer to our intended purpose—the set of sentences in a given language.

There is also the empty set, \( \emptyset \). This is the one set with no members. That is, for each object \( u \), \( u \) is not a member of \( \emptyset \) (i.e.: for each \( u, u \notin \emptyset \).)

Though the notion of a set is an intuitive one, it is deeply perplexing. This can be seen by reflecting on the Russell Paradox, discovered by Bertrand Russell, the great philosopher and mathematician. Let us call \( R \) the set of all and only

---

5 Supplementary reading: the beginning of Enderton (1977)
those sets that are not members of themselves. For short, \( R \) is the set of non-self-members. Russell asks the following question: is \( R \) a member of itself? There are two possibilities:

i) \( R \notin R \). Thus, \( R \) is a non-self-member. But \( R \) was said to be the set of all non-self-members, and so we’d have \( R \in R \).

Contradiction.

ii) \( R \in R \). So \( R \) is not a non-self-member. \( R \), by definition, contains only non-self-members. So \( R \notin R \). Contradiction.

Thus, each possibility leads to a contradiction. But there are no remaining possibilities—either \( R \) is a member of itself or it isn’t! So it looks like the very idea of sets is paradoxical.

The modern discipline of axiomatic set theory arose in part to develop a notion of sets that isn’t subject to this sort of paradox. This is done by imposing rigid restrictions on when a given “condition” picks out a set. In the example above, the condition “is a non-self-member” will be ruled out—there’s no set of all and only the things satisfying this condition. The details of set theory are beyond the scope of this course; for our purposes, we’ll help ourselves to the existence of sets, and not worry about exactly what sets are, or how the Russell paradox is avoided.

Various other useful set-theoretic notions can be defined in terms of the notion of membership. We say that \( A \) is a subset of \( B \) (“\( A \subseteq B \)””) when every member of \( A \) is a member of \( B \). We say that the intersection of \( A \) and \( B \) (“\( A \cap B \)””) is the set that contains all and only those things that are in both \( A \) and \( B \), and that the union of \( A \) and \( B \) (“\( A \cup B \)””) is the set containing all and only those things that are members of either \( A \) or \( B \).

Suppose we want to refer to the set of the so-and-sos—that is, the set containing all and only objects, \( u \), that satisfy the condition “so-and-so”. We’ll do this with the term “[\( u : u \) is a so-and-so]”. Thus, we could write: “\( \mathbb{N} = \{ u : u \) is a natural number\}”. And we could restate the definitions of \( \cap \) and \( \cup \) from the previous paragraph as follows:

\[
A \cap B =_{df} \{u : u \in A \text{ and } u \in B\}
\]

\[
A \cup B =_{df} \{u : u \in A \text{ or } u \in B\}
\]

Sets have members, but they don’t contain them in any particular order. For example, the set containing me and Bill Clinton doesn’t have a “first” member. This is reflected in the fact that “[Ted, Clinton]” and “[Clinton, Ted]” are
two different names for the same set—the set containing just Clinton and Ted. But sometimes we need to talk about a set-like thing containing Clinton and Ted, but in a certain order. For this purpose, logicians use ordered sets. Two-membered ordered sets are called ordered pairs. To name the ordered pair of Clinton and Ted, we use: “(Clinton, Ted)”. Here, the order is significant, for (Clinton, Ted) and (Ted, Clinton) are not the same thing. The three-membered ordered set of \(u, v, \) and \(w\) (in that order) is written: \((u, v, w)\); and similarly for ordered sets of any finite size. A \(n\)-membered ordered set is called an \(n\)-tuple. Let’s even allow 1-tuples: let’s define the 1-tuple \(\langle u \rangle\) as being the object \(u\) itself.

In addition to sets, and ordered sets, we’ll need a further related concept: that of a function. A function is a rule that “takes in” an object or objects, and “spits out” a further object. For example, the addition function is a rule that takes in two numbers, and spits out their sum. As with sets and ordered sets, functions are not limited to mathematical entities: they can “take in” and “spit out” any objects whatsoever. We can speak of the father-of function, for example, which is a rule that takes in a person, and spits out the father of that person. And later in this book we will be considering functions that take in and spit out linguistic entities: sentences and parts of sentences from formal languages.

Each function has a fixed number of “places”: a fixed number of objects it must take in before it is ready to spit out something. You need to give the addition function two arguments (numbers) in order to get it to spit out something, so it is called a two-place function. You only need to give the father-of function one object, on the other hand, to get it to spit out something, so it is a one-place function.

The objects that the function takes in are called its arguments, and the object it spits out is called its value. Suppose \(f\) is an \(n\)-place function, and \(u_1\ldots u_n\) are \(n\) of its arguments; one then writes “\(f(u_1\ldots u_n)\)” for the value of function \(f\) as applied to arguments \(u_1\ldots u_n\). \(f(u_1\ldots u_n)\) is the object that \(f\) spits out, if you feed it \(u_1\ldots u_n\). For example, where \(f\) is the father-of function, since Ron is my father, we can write: \(f(\text{Ted}) = \text{Ron}\); and, where \(a\) is the addition function, we can write: \(a(2,3) = 5\).

There’s a trick for “reducing” talk of both ordered pairs and functions to talk of sets. One first defines \(\{u, v\}\) as the set \(\{\{u\}, \{u, v\}\}\); one defines \(\{u, v, w\}\) as \(\{u, \langle v, w \rangle \}\), and similarly for \(n\)-membered ordered sets, for each positive integer \(n\). And, finally, one defines an \(n\)-place function as a set, \(f\), of \(n+1\)-tuples obeying the constraint that if \(\langle u_1, \ldots, u_n, v \rangle\) and \(\langle u_1, \ldots, u_n, w \rangle\) are both members of \(f\), then \(v = w\); \(f(u_1, \ldots, u_n)\) is then defined as the object, \(v\), such
that \( (u_1, \ldots, u_n, v) \in f \). Thus, ordered sets and functions are defined as certain sorts of sets. The trick of the definition of ordered pairs is that we put the set together in such a way that we can look at the set and tell what the first member of the ordered pair is: it's the one that “appears twice”. Similarly, the trick of the definition of a function is that we can take any arguments to the function, look at the set that is identified with the function, and figure out what value the function spits out for those arguments. But the technicalities of these reductions won’t matter for us; I’ll just feel free to speak of ordered pairs, triples, functions, etc., without defining them as sets.