Metatheory: The Basic Concepts

10

10–1. OBJECT LANGUAGE AND METALANGUAGE

In metatheory, we analyze and prove facts **about** logic, as opposed to **using** logic. To proceed clearly, we must bear in mind that the language in which we do logic is distinct from the language in which we study logic—that is, that the language of sentence and predicate logic is distinct from English. The distinction has been implicit throughout the text. It is time to make this distinction explicit and give it a name.

Since the language of sentence and predicate logic is the language we study and talk about, we call it an *Object language*.

> An *Object Language* is a language we study and talk abut. Our object language is the language of sentence and predicate logic.

Our object language has an infinite stock of sentence letters, names, one place predicates, two place predicates, and in general, n-place predicates. (In section 15–5 we also add function symbols.)

We contrast our object language with the language, called a *Metalanguage,* which we use to talk about our object language. Our metalanguage is English, to which we add a few convenient items. Most of these you have already seen. For example, think about how we have been using boldface capital '**X**' and '**Y**' to range over sentences in the object language. In so doing, we are really using '**X**' and '**Y**' as a simple extension of English, as a new linguistic device which we use in talking about the language of sentence and predicate logic. We have used '**s**', '**t**', '**u**', '**v**', '**P(u)**', and '**R(u,v)**' similarly. Since these are variables used in the metalanguage to range over object language sentences, names, variables, and open sentences, we call them *Metavariables.*

I will now add three more kinds of metavariables to be used as part of English in talking about our object language. I will use boldface script capitals '***X***', '***Y***', and '***Z***' to talk generally about **sets** of object language sentences. A set of sentences is just a collection of one, two, or more sentences where the order of the sentences does not matter. I will also include among sets of sentences infinite sets, with infinitely many sentences in them, and the somewhat funny case of the *Empty Set,* that is, the degenerate case in which the set contains nothing.

Next, I will use '**I**', '**J**', . . . as metavariables ranging over interpretations. When, as in chapter 15, we will be concerned with predicate logic sentences, interpretations will be described by a generalization of the idea I introduced in chapter 2. For chapters 11 to 14, in which we will be concerned only with sentence logic, interpretations will just be assignments of truth values to atomic sentence letters, that is, specifications of conditions which determine truth values for sentence logic sentences. I will use '**I**' and '**J**' as part of English to talk generally about interpretations, as when I say that two sentences, **X** and **Y**, are logically equivalent if and only if, for each **I** which is an interpretation of both **X** and **Y**, either **X** and **Y** are both true in **I** or **X** and **Y** are both false in **I**.

As a last metavariable I will use '**T**' to range over truth trees.

I will also add to English the special symbol '\' as an abbreviation of the word 'Therefore'. ***Z*\X** stands for the argument which has sentences in the set ***Z*** as its premises and **X** as its conclusion. This is exactly what I have previously written as "***Z***. Therefore **X**." I did not previously introduce '\' as an abbreviation for 'therefore' because I wanted to be sure that you did not mistakenly think that '\' was part of the object language. But now that we have made the object language/metalanguage distinction explicit, I can introduce '\' as an abbreviation for 'therefore' and ask you to be careful to understand that '\' is an abbreviation in English, not a connective of the object language we are studying. To summarize

> A *Metalanguage* is a language, distinct from the object language, which we use to talk about the object language. Our metalanguage is English, augmented with metavariables as follows: '**X**', '**Y**', '**Z**', . . . range over object language sentences; '***X***', '***Y***', '***Z***', . . . range over sets of object language sentences*; '**s**', '**t**', . . . range over names in the object language; '**u**', '**v**', . . . range over variables of the object language; '**P(u)**' and '**R(u, v)**' . . . range over sentences of the object language in which **u** (or **u** and **v**) may be free; '**I**', '**J**', . . . range over interpretations; and '**T**' ranges over trees. We also use '\' as an abbreviation for 'therefore' in the metalanguage, so that '***Z*\X**' stands for the argument with premises taken from the sentences in the set ***Z*** and conclusion the sentence **X**.

*Only after typesetting made large-scale changes in type a practical impossibility, I learned that the compositor's capital boldface italic was almost indistinguishable from the roman boldface type. However, I have used ***Z*** everywhere as my metavariable for sets of sentences, with only two minor exceptions (where I use ***W***); and **Z** never occurs as a metavariable for sentences. By remembering that ***Z*** ranges over sets of sentences, I hope that the reader will be able to make the needed contrast. I regret not having provided a truly distinctive typeface.

To understand better the interplay between object and metalanguage, you also need to understand the distinction between *Use* and *Mention.* Let's talk for a moment about Adam: In so doing I mention (that is, I refer to) this **person**. I might say about Adam that he is blond. Now, let us talk, not about the person, Adam, but about this person's **name**, 'Adam'. For example, I might say that 'Adam' is spelled with four letters. Note how I accomplished this. To talk about the name, I take the name and enclose it in single quotation marks. If I use the name without quotes, I use the name to mention (that is, to talk about) the person. If I use the name enclosed in quotes, I use the quoted name—really a name of the name—to mention (talk about) the name.

Throughout this text I have tried hard (but not always successfully!) to observe the distinction between use and mention. Thus, when in the text I have talked about an object language sentence, such as 'A&B', I have been careful always to enclose it in quotes. When such a sentence is displayed as an example, like this

A&B

I omit the quotes. This is because of the convention, universal in logic and philosophy, that offsetting a formal expression functions just like quoting it, so that you know that we are talking about what has been displayed rather than using what is displayed to make a statement or reference.

In contrast, when I use a metavariable I do not put quotes around it. Thus I might say that if the sentence **X** is a conjunction, then **X** contains the symbol '&'. Notice that there are no quotes around the boldface letter. This is because I was **using** it to make a general statement, not mentioning the letter. In contrast, I do use quotes when I **mention** (that is, talk about) the boldface letter, as in the following statement: In the previous example I used the symbol '**X**' as an example of how metavariables can be used.

Now let's look at a problematic case. Suppose I say that any sentence of the form **X&Y** is true just in case **X** and **Y** are both true. I have, writing in the metalanguage, used '**X**' and '**Y**' to make a general statement. But in so doing I used the expression '**X&Y**', which contains the object language symbol '&'. Furthermore, in some sense I made a statement **about** the symbol '&'. I didn't assert a conjunction. Instead, I talked about all sentences which have '&' as their main connective.

Here's the problem. I was tacitly talking **about** the symbol '&'. But I didn't quote it. I really should have used quotes around '&'. But it's not clear how I could do that without putting quotes around '**X**' and '**Y**', which I was using and not mentioning!

Philosophers have invented some fancy notation to make more precise what is going on in such cases. But introducing this further notation would be to pass the point of diminishing returns for our present needs. Instead, I am simply going to ask you to understand that such "mixed" cases of use and mention, formed with metalanguage boldface variables and object language connectives, are a device which I use to talk generally about all sentences of the indicated form.

I must mention one further twist in our conventions. Our object language provides a very precise and compact way of expressing truth functional facts. It would be a shame not to be able to use this compact notation in our metalanguage and to have to write everything out in imprecise, long-winded English. So we will occasionally allow ourselves the luxury of using expressions of the object language to make statements as part of the metalanguage. You can think of the metalanguage, English, as incorporating or being extended by a copy of the object language.

You can always tell when I talk about, or mention, logical notation as part of the object language, for in these cases I will always quote or display the expressions. When I use, as opposed to mention, logical notation as part of the metalanguage, the notation will not be quoted. Furthermore, when I use, as opposed to mention, logical notation as part of the metalanguage, I will use the notation with metalanguage variables. You can spot these metalanguage variables as belonging to the metalanguage because I always write them in boldface. Strictly speaking, my notation does not distinguish between use of logical notation in the metalanguage

EXERCISES

10–1. For each of the underlined expressions, say whether the expression is being used as part of the metalanguage, mentioned as part of the metalanguage, used as part of the object language, or mentioned as part of the object language.

a) If there is a proof of a sentence **X**, then there is a proof of the sentence **X**v**Y**.
b) The sentence '(∀x)(Bx v ~Bx)' is a logical truth.
c) Any sentence of the form **(∀u)[P(u) v ~P(u)]** is a logical truth.
d) '**Y**' is a metavariable.

and the mixed use-mention cases which I described two paragraphs back. But in practice this imprecision causes no confusion.

10–2. SYNTAX AND SEMANTICS

Much of metatheory deals with connections between syntax and semantics, another distinction which I have tacitly observed throughout the text. A fact of *Syntax* is a fact which concerns symbols or sentences insofar as the fact can be determined from the **form** of the symbols or sentences, from the way they are written. The point is that facts of syntax do not depend on what the symbols mean.

A fact of *Semantics*, on the other hand, concerns the referents, interpretation, or (insofar as we understand this notion) the meaning of symbols and sentences. In particular, semantics has to do with the referents of expressions, the truth of sentences, and the relations between various cases of these.

Here are some examples: Syntactic facts include the fact that 'A&B' is a well-formed sentence of sentence logic, that 'AB&' is not a well-formed sentence, and that 'A&B' is a conjunction. Syntactic facts include more general facts which can be determined from form alone, such as the fact that the derivation rule &E and the truth tree rule & apply to any sentence of the form **X&Y** and that any sentence of the form **X**v**Y** is derivable from (that is, there is a proof from) a sentence of the form ~**X** ⊃ **Y**.

One thing to keep in mind is that whether or not a given string of sentences counts as a formal proof (a derivation or a tree) is a syntactic fact. All the rules of proof have been carefully stated so that they appeal only to facts about how sentences are written, not about how they are interpreted. Whether or not a string of sentences qualifies as a proof depends only on the form, not on the content of the sentences. To see this clearly, consider that you could program a computer to check and construct proofs. The computer need not know **anything** about how the sentences are interpreted. For example, the computer need not know that you are supposed to think of '&' as meaning 'and'. It need only be programmed so that if a sentence of the form **X&Y** appears on a derivation or tree, then below this sentence it can write both the sentences **X** and **Y**.

Examples of semantic facts include the fact that any interpretation which makes 'A⊃B' true makes '~B⊃~A' true, that '(∀x)(Px v ~Px)' is true in all interpretations, and that '(∀x)Px' is true in some interpretations and false in others. Semantic facts include more general facts such as the fact that any existentially quantified sentence is true in an interpretation if one of its substitution instances is true in the interpretation.

To summarize the distinction between syntactic and semantic facts

Facts of *Syntax* are facts having to do with the form of expressions. Syntactic facts contrast with facts of *Semantics* which have to do with the truth, reference, and the meaning of expressions.

EXERCISES

10–2. Which of the following facts are syntactic facts and which semantic facts?

a) Any interpretation which makes '(∀x)(Ax & Bx)' true makes '(∀x)Ax' true
b) The expression 'A&BvC' is not a well-formed sentence, though it would be if parentheses were put around the 'A&B'.
c) A sentence of the form ~~**X** can be derived from a sentence of the form **X**.
d) In some interpretations 'a' and 'b' have the same referent. In some interpretations they do not.
e) If **X** and **Y** are well-formed sentences, then so is their conjunction.
f) If the argument **X****Y** is valid, then so is the argument ~**Y**\\~**X**.
g) A model of a set of sentences (that is, an interpretation in which each sentence in the set is true) is a model for any subset of the set (that is, any smaller set of sentences all the sentences of which are contained in the original set).
h) If there is a proof of the sentence **X** from the sentences in the set **Z**, then there is a proof of **X** from any superset of **Z**, that is, any set which contains all the sentences of **Z** as well as one or more additional sentences.

10–3. SOUNDNESS AND COMPLETENESS

Students often have difficulty appreciating the difference between the question of whether an argument, **Z****X**, is valid (a semantic question) and the question of whether there is a proof from **Z** to **X** (a syntactic question). And no wonder! The syntactic rules of proof have been carefully crafted so that there is a proof from **Z** to **X** if and only if the argument, **Z****X**, is valid. Of course, we have done this so that we can use proofs to ascertain validity. But this must not obscure the fact that *Derivability*—that is, the existence of a proof—is one thing and validity is another. That these two very different concepts go together is something we must demonstrate. Indeed, this fundamental result about logic is what the rest of this book is about.

To help in talking about these ideas, we will use two new abbreviations in the metalanguage. (The following definitions also use the abbreviation 'iff', which is just shorthand for the metalanguage expression 'if and only if'.)

D1: **Z**⊢**X** iff **X** is *Derivable* from **Z**, that is, iff there is a formal proof of **X** using only sentences in **Z**.

D2: **Z**⊨**X** iff the argument **Z****X** is valid, that is, iff every interpretation which makes all of the sentences in **Z** true also makes **X** true.

The symbol '⊢' is called the *Single Turnstyle*. **Z**⊢**X** asserts that a **syntactic** relation holds between the sentences in the set of sentences, **Z**, and the sentence, **X**, that the latter is derivable from the former. The symbol '⊨' is called the *Double Turnstyle*. **Z**⊨**X** asserts that a **semantic** relation holds between the set of sentences, **Z**, and the sentence, **X**, that any interpretation which makes all of the sentences in **Z** true will also make **X** true.

Here's a mnemonic to help remember which turnstyle is which. '⊨' has ***m***ore bars and so has to do with ***m***eaning. '⊢' has ***l***ess bars and so has to do with the form of ***l***anguage.

Using the turnstyle notation, we can express the close relation between derivability and validity in two convenient parts:

D3: A system of formal proof is *Sound* iff for all **Z**, **X**, if **Z**⊢**X**, then **Z**⊨**X**.

To say that a system of formal proof is sound is to say that whenever you have a proof from **Z** to **X**, then any interpretation which makes all of the sentences in **Z** true also makes **X** true.

D4: A system of formal proof is *Complete* iff for all **Z**, **X**, if **Z**⊨**X**, then **Z**⊢**X**.

To say that a system of formal proof is complete is to say that in every case of an argument, **Z****X**, which is valid (that is, any interpretation which makes every sentence in **Z** true also makes **X** true), there exists a proof from **Z** to **X**. Completeness means that there is a proof in every case in which there ought to be a proof.

Once more, derivability and validity are distinct concepts. But derivability has been set up so that it can be used as a surefire test of validity. To give a crude analogy, derivability is like the litmus test for acids. If you put a piece of litmus paper in a liquid and the paper turns red, you know that the liquid is an acid. If the litmus paper does not turn red, the liquid is not an acid. Derivability is a kind of litmus test for validity. Proving that the test works, proving soundness and completeness, is the fundamental metatheoretical result in logic.

This litmus test analogy is a good way to emphasize the fact that derivability and validity are distinct but related ideas. However, I must be sure that the analogy does not mislead you in the following respect. Derivability is a surefire test for validity in the sense that if there is a proof, then the corresponding argument is valid, and if an argument is valid, then there exists a proof which establishes that validity. But there may not be any surefire way to establish whether or not such a proof exists! We might look for a proof from **Z** to **X** until the cows come home and still not know for sure whether or not a proof exists.

In predicate logic there is no mechanical means to determine whether or not a proof from **Z** to **X** exists, no means guaranteed to give a definite yes or no answer in some finite number of steps. This fact about predicate logic is known as *Undecidability*, and constitutes a second fundamental metatheoretical result. (Sentence logic is decidable.) If you learned the tree method, I can give you a hint of what is involved by reminding you of the problem of infinite trees. The same fact will turn up for derivations when we get to chapter 15. However, further study of undecidability goes beyond what you will study in this text.

EXERCISES

10–3. Some one might propose a set of rules of inference different from our natural deduction or truth tree rules. Explain what is involved in such a new set of rules being *Unsound* (not sound) or *Incomplete* (not complete).

In fact, logicians have proposed many, many sets of inferential rules. Some such sets are sound and complete, some are not. Whenever someone proposes a new set of inference rules it is important to determine whether or not the rules are sound and complete.

Exercises 10–4 to 10–6 concern the idea of *Rule Soundness*. To say that an individual rule of inference is sound is to say that if the rule is applied to a sentence or sentences which is (are) true in a case, then the sentence which the rule licenses you to draw is also true in that case. We can state the rules of inference for derivations using the turnstyle notation, and we can also use this notation to assert the soundness of these rules. For example, the rule &I is expressed by saying that if **Z**⊢**X** and **Z**⊢**Y**, then **Z**⊢**X&Y**. We can state, in one way, that the rule &I is sound by stating that if **Z**⊨**X** and **Z**⊨**Y**, then **Z**⊨**X&Y.**

10–4. Show that the rule &I is sound.

10–5. State the other primitive rules for derivations using the turnstyle notation and show that they are sound.

10–6. Consider the following new rules for derivations. Determine which are sound and which are not. In each case, give an informal demonstration of your conclusion about the rules.

a) If **Z**⊢**X**⊃**Y** and **Z**⊢**Y**, then **Z**⊢**X**.
b) If **Z**⊢**X**≡**Y** and **Z**⊢~**X**, then ⊢~**Y**.
c) If **Z**⊢[(∀u)**P(u)** v (∀u)**Q(u)**], then **Z**⊢(∀u)[**P(u)** v **Q(u)**].
d) If **Z**⊢(∃u)**P(u)**, then **Z**⊢**P(s)**.
e) If **Z**⊢[(∃u)**P(u)** & (∃u)**Q(u)**], then **Z**⊢(∃u)[**P(u)** & **Q(u)**].

10–7. Refresh your memory of the truth table method of establishing validity in sentence logic (see exercise 4–2 in chapter 4 of volume I). Then show that this method provides a decision procedure for sentence logic. That is, show that, given a sentence logic argument, the truth table method is guaranteed in a finite number of steps to give you a yes or no answer to the question of whether or not the argument is valid.

10–4. SOME FURTHER NOTATION AND DEFINITIONS

Some further notation and definitions will prove very useful in the following chapters, and will also give you a chance to practice the concepts of the last three sections.

First, here's an obvious and trivial extension of the turnstyle notation, a fussy logician's point which you might not even notice. For example, if I write '**Z**⊢**X**', I have used **Z** as a metavariable over **sets** of sentences. What if I want to look at the special case in which **Z** contains just one sentence? Then I may just use '**Z**', a metavariable over individual sentences, writing '**Z**⊢**X**'. Or, if I want more explicitly to list the sentences that come before the turnstyle, I may do just that, explicitly giving the list, for example, writing **W,Z**⊢**X.** I may use the same latitude in notation with the double turnstyle.

A little less trivially, I have glossed over an important point about using the single turnstyle. '**Z**⊢**X**' means that there is a proof of **X** from the sentences in the set **Z**. But by proof, do I mean a derivation or a closed tree? It is important to keep separate these very distinct kinds of formal proof. Strictly speaking, I should use one kind of turnstyle, say, '$\vdash_d$' to mean derivations. Thus '**Z**$\vdash_d$**X**' means that there is a derivation which uses premises in **Z** and has **X** as its last line. And I should use a second kind of turnstyle, say, '$\vdash_t$', to mean trees. Thus '**Z**$\vdash_t$**X**' means that there is a closed tree of which the initial sentences are ~**X** and sentences taken from **Z**. Other systems of formal proof (and there are **many** others) must be distinguished from these with further subscripts on the single turnstyle. When there is any danger of confusion about what kind of proof is in question, we must use a disambiguating subscript on the turnstyle. Usually, context will make it quite plain which kind of proof is in question, In which case we may omit the subscript.

EXERCISE

10–8. Do we need corresponding subscripts on the double turnstyle? Explain why or why not.

Here is one more refinement. How should we understand the turnstyle notation when the set Z has infinitely many sentences? In the case of '$\mathbf{Z}\vDash\mathbf{X}$' this should be clear enough. This asserts that every interpretation which makes **all** of the infinitely many sentences in **Z** true also makes **X** true. But what do we mean by '$\mathbf{Z}\vdash\mathbf{X}$'? A formal proof can use only finitely many sentences. So by '$\mathbf{Z}\vdash\mathbf{X}$' we mean that there is a proof of **X** each premise of which is a sentence in the set **Z**. This formulation leaves it open whether all of the sentences in **Z** get used as premises. If **Z** is infinite, only finitely many sentences can be used in a proof from **Z**. If **Z** is finite, all or only some of the sentences in **Z** may get used. Reread definition D1 and be sure that you understand 'formal proof of **X using only sentences in Z**', as just explained, to mean a proof which uses any number of, but not necessarily all, the sentences in **Z** as premises. We even allow the case of using no premises at all. Any proof of a sentence, **X**, from no premises makes $\mathbf{Z}\vdash\mathbf{X}$ true for any set of sentences, **Z**.

EXERCISES

10–9. '$\mathbf{Z}\subset\mathbf{W}$' means that every sentence in **Z** is a sentence in **W**. We say that **Z** is a *Subset of* **W**. Show that if $\mathbf{Z}\vdash\mathbf{X}$ and $\mathbf{Z}\subset\mathbf{W}$, then $\mathbf{W}\vdash\mathbf{X}$.

10–10. Show that if $\mathbf{Z}\vDash\mathbf{X}$ and $\mathbf{Z}\subset\mathbf{W}$, then $\mathbf{W}\vDash\mathbf{X}$.

10–11. If **Z** is the empty set, we write $\vdash\mathbf{X}$ for $\mathbf{Z}\vdash\mathbf{X}$ and $\vDash\mathbf{X}$ for $\mathbf{Z}\vDash\mathbf{X}$. Explain what $\vdash\mathbf{X}$ and $\vDash\mathbf{X}$ mean.

If you have studied truth trees, you have already encountered (in section 9–2, volume I, and section 8–1, this volume) the idea of a *Model* of a set of sentences. It's not complicated: An interpretation, **I**, is a model of a set of sentences, **Z**, iff every sentence in **Z** is true in **I**. That is, a model, **I**, of a set of sentences, **Z**, makes all the sentences in **Z** true. For example, consider the truth value assignment, **I** which makes 'A' true, 'B' false, and 'C' true. **I** is a model for the set of sentences {(AvB),(~B⊃C)}, but is not a model for the set of sentences {(A&~B),C,(B≡C)}. Be sure you understand why this is so. To check, work out the truth values of each sentence in the two sets in the truth value assignment, **I**, and apply the definition of model just given.

In the following chapter we will use the notion of a model so often that it's worth introducing an abbreviation:

D5: '**Mod**' is a predicate in the metalanguage (an abbreviation in English), defined as **Mod(I,Z)** iff all the sentences in the set **Z** are true in the interpretation, **I**. If **Mod(I,Z)**, **I** is said to be a *Model* for the sentences in **Z**. **I** is also said to *Satisfy* the sentences in **Z**.

As with the turnstyle notation, we can use metavariables for sentences, such as '**Z**', where the metavariable, '**Z**', for sets of sentences occurs in the definition of '**Mod**'.

We will also lean heavily on the notations of consistency and inconsistency, already introduced in exercise 7–8 and section 9–2 (in volume I) and in sections 6–3 and 8–1 (in this volume). To get ready for this work, and to practice this chapter's ideas, here is a pair of equivalent definitions for each of these concepts. (The slash through the double turnstyle in D6′ means just what a slash through an equal sign means—the double turnstyle relation does **not** hold.)

D6: The set **Z** of sentences is *Consistent* iff (∃**I**)**Mod(I,Z)**.

D6′: The set **Z** of sentences is *Consistent* iff $\mathbf{Z}\nvDash$A&~A.

D7: The set *Z* of sentences is *Inconsistent* iff (∀**I**)~**Mod(I,Z)**, that is, iff **Z** is not consistent, that is, iff there is no model for all the sentences in **Z**.

D7′: The set *Z* of sentences is *Inconsistent* iff $\mathbf{Z}\vDash$A&~A.

EXERCISES

10–12. Show that D6 and D6′ are equivalent.

10–13. Show that D7 and D7′ are equivalent.

10–14. Explain why the notions of consistency and inconsistency are semantic and not syntactic notions. Modify definitions D6′ and D7′ to provide corresponding syntactic notions, and label your new definitions D6″ and D7″. You will then have a pair of notions, *Semantic Consistency* and *Syntactic Consistency*, and a second pair, *Semantic Inconsistency* and *Syntactic Inconsistency*. You must always carefully distinguish between these semantic and syntactic ideas. Whenever I speak about consistency and inconsistency without specifying whether it is the semantic or syntactic notion, I will always mean the semantic notion.

10–15. What do you think the relation is between semantic and syntactic consistency, and between semantic and syntactic inconsistency? What would guess is the connection between this question and the ideas of soundness and completeness? Write a paragraph informally explaining these connections as best you can.

CHAPTER CONCEPTS

Here are the important concepts which I have introduced and discussed in this chapter. Review them carefully to be sure you understand them.

a) Object Language
b) Metalanguage
c) Metavariable
d) Use
e) Mention
f) Syntactic Fact
g) Semantic Fact
h) Derivability
i) $\vdash$
j) $\vDash$
k) Soundness
l) Completeness
m) Set of sentences
n) Subset
o) Model
p) Consistency
q) Inconsistency