A Modern Formal Logic Primer

Volume II

Predicate Logic 1

Syntax

1-1. WE NEED MORE LOGICAL FORM

In Volume I you gained a firm foundation in sentence logic. But there must be more to logic, as you can see from the next examples. Consider the following two English arguments and their transcriptions into sentence logic:

(1)	Everyone loves Adam.	A	(2)	Eve loves Adam.	B
	Eve loves Adam.	B		Someone loves Adam.	C

In sentence logic, we can only transcribe the sentences in these arguments as atomic sentence letters. But represented with sentence letters, both natural deduction and truth trees tell us that these arguments are invalid. No derivation will allow us to derive 'B' from 'A' or 'C' from 'B'. A&~B is a counterexample to the first argument, and B&~C is a counterexample to the second. An argument is valid only if it has no counterexamples.

Something has gone terribly wrong. Clearly, if everyone loves Adam, then so does Eve. If the premise is true, without fail the conclusion will be true also. In the same way, if Eve loves Adam, then someone loves Adam. Once again, there is no way in which the premise could be true and the conclusion false. But to say that if the premises are true, then without fail the conclusion will be true also is just what we intend when we say that an argument is valid. Since sentence logic describes these arguments as invalid, it looks like something has to be wrong with sentence logic.

Sentence logic is fine as far as it goes. The trouble is that it does not go far enough. These two arguments owe their validity to the internal logical structure of the sentences appearing in the arguments, and sentence logic does not describe this internal logical structure. To deal with this shortcoming, we must extend sentence logic in a way which will display the needed logical structure and show how to use this structure in testing arguments for validity. We will keep the sentence logic we have learned in Volume I. But we will extend it to what logicians call *Predicate Logic* (also sometimes called *Quantificational Logic*).

Predicate logic deals with sentences which say something about someone or something. Consider the sentence 'Adam is blond.' This sentence attributes the property of being blond to the person named 'Adam'. The sentence does this by applying the predicate (the word) 'blond' to the name 'Adam'. A sentence of predicate logic does the same thing but in a simplified way.

We will put capital letters to a new use. Let us use the capital letter 'B', not now as a sentence letter, but to transcribe the English word 'blond'. And let us use 'a' to transcribe the name 'Adam'. For 'Adam is blond.', predicate logic simply writes 'Ba', which you should understand as the predicate 'B' being applied to the name 'a'. This, in turn, you should understand as stating that the person named by 'a' (namely, Adam) has the property indicated by 'B' (namely, the property of being blond).

Of course, on a different occasion, we could use 'B' to transcribe a different English predicate, such as 'bachelor', 'short', or 'funny'. And we could use 'a' as a name for different people or things. It is only important to stick to the same transcription use throughout one problem or example.

Predicate logic can also express relations which hold between things or people. Let's consider the simple statement that Eve loves Adam. This tells us that there is something holding true of Eve and Adam together, namely, that the first loves the second. To express this in predicate logic we will again use our name for Adam, 'a'. We will use a name for Eve, say, the letter 'e'. And we will need a capital letter to stand for the relation of loving, say, the letter 'L'. Predicate logic writes the sentence 'Eve loves Adam.' as 'Lea'. This is to be read as saying that the relation indicated by 'L' holds between the two things named by the lowercase letters 'e' and 'a'. Once again, in a different example or problem, 'L', 'a', and 'e' could be used for different relations, people, or things.

You might be a little surprised by the order in which the letters occur in 'Lea'. But don't let that bother you. It's just the convention most often used in logic: To write a sentence which says that a relation holds between two things, first write the letter which indicates the relation and then write the names of the things between which the relation is supposed to hold. Some logicians write 'Lea' as 'L(e,a)', but we will not use this notation.

Note, also, the order in which the names 'e' and 'a' appear in 'Lea'. 'Lea' is a different sentence from 'Lae'. 'Lea' says that Eve loves Adam. 'Lae' says that Adam loves Eve. One of these sentences might be true while the other one is false! Think of 'L' as expressing the relation, which holds just in case the **first** thing named loves the **second** thing named.

Here is a nasty piece of terminology which I have to give you because it is traditional and you will run into it if you continue your study of logic. Logicians use the word *Argument* for a letter which occurs after a predicate or a relation symbol. The letter 'a' in 'Ba' is the argument of the predicate 'B'. The letters 'e' and 'a' in 'Lea' are the arguments of the relation symbol 'L'. This use of the word 'argument' has nothing to do with the use in which we talk about an argument from premises to a conclusion.

At this point you might be perplexed by the following question. I have now used capital letters for three different things. I have used them to indicate atomic sentences. I have used them as predicates. And I have used them as relation symbols. Suppose you encounter a capital letter in a sentence of predicate logic. How are you supposed to know whether it is an atomic sentence letter, a predicate, or a relation symbol?

Easy. If the capital letter is followed by two lowercase letters, as in 'Lea', you know the capital letter is a relation symbol. If the capital letter is followed by one lowercase letter, as in 'Ba', you know the capital letter is a predicate. And if the capital letter is followed by no lowercase letters at all, as in 'A', you know it is an atomic sentence letter.

There is an advantage to listing the arguments of a relation symbol after the relation symbol, as in 'Lea'. We can see that there is something important in common between relation symbols and predicates. To attribute a relation as holding between two things is to say that something is true about the two things taken together and in the order specified. To attribute a property as holding of one thing is to say that something is true about that one thing. In the one case we attribute something to one thing, and in the other we attribute something to two things.

We can call attention to this similarity between predicates and relations in a way which also makes our terminology a bit smoother. We can indicate the connection by calling a relation symbol a *Two Place Predicate*, that is, a symbol which is very like an ordinary predicate except that it has two argument places instead of one. In fact, we may sometimes want to talk about **three** place predicates (equally well called 'three place relation symbols'). For example, to transcribe 'Eve is between Adam and Cid', I introduce 'c' as a name for Cid and the three place predicate 'K' to indicate the three place relation of being between. My transcription is 'Keac', which you can think of as saying that the three place relation of being between holds among Eve, Adam, and Cid, with the first being between the second and the third.

This is why our new logic is called 'predicate logic': It involves predicates of one place, two places, three places, or indeed, any number of places. As I mentioned, logicians also refer to these symbols as one place, two place, or many place relation symbols. But logicians never call the resulting system of logic 'relation logic'. I have no idea why not.

Our familiar sentence logic built up all sentences from atomic sentence letters. Predicate logic likewise builds up compound sentences from atomic sentences. But we have expanded our list of what counts as an atomic sentence. In addition to atomic sentence letters, we will include sentences such as 'Ba' and 'Lea'. Indeed, any one place predicate followed by one name, any two place predicate followed by two names, and so on, will now also count as an atomic sentence. We can use our expanded stock of atomic sentences to build up compound sentences with the help of the connectives, just as before.

How would you say, for example, 'Either Eve loves Adam or Adam is not blond.'? 'Lea v ~Ba'. Try 'Adam loves himself and if he is blond then he loves Eve too.': 'Laa & (Ba ⊃ Lae)'.

In summarizing this section, we say

> In predicate logic, a capital letter without a following lowercase letter is (as in sentence logic) an atomic sentence. Predicate logic also includes predicates applied to names among its atomic sentences. A capital letter followed by one name is a *One Place Predicate* applied to one name. A capital letter followed by two names is a *Two Place Predicate* applied to two names, where the order of the names is important. Predicates with three or more places are used similarly.

EXERCISES

In the following exercises, use this transcription guide:

a: Adam
e: Eve
c: Cid
Bx: x is blond
Cx: x is a cat
Lxy: x loves y
Txy: x is taller than y

1–1. Transcribe the following predicate logic sentences into English:

a) Tce
b) Lce
c) ~Tcc
d) Bc
e) Tce ⊃ Lce
f) Lce v Lcc
g) ~(Lce & Lca)
h) Bc ≡ (Lce v Lcc)

1–2. Transcribe the following English sentences into sentences of predicate logic;

a) Cid is a cat.
b) Cid is taller than Adam.
c) Either Cid is a cat or he is taller than Adam.
d) If Cid is taller than Eve then he loves her.
e) Cid loves Eve if he is taller than she is.
f) Eve loves both Adam and Cid.
g) Eve loves either Adam or Cid.
h) Either Adam loves Eve or Eve loves Adam, but both love Cid.
i) Only if Cid is a cat does Eve love him.
j) Eve is taller than but does not love Cid.

1–2. QUANTIFIERS AND VARIABLES

We still have not done enough to deal with arguments (1) and (2). The sentences in these arguments not only attribute properties and relations to things, but they involve a certain kind of generality. We need to be able to express this generality, and we must be careful to do it in a way which will make the relevant logical form quite clear. This involves a way of writing general sentences which seems very awkward from the point of view of English. But you will see how smoothly everything works when we begin proving the validity of arguments.

English has two ways of expressing general statements. We can say 'Everyone loves Adam.' (Throughout, 'everybody' would do as well as 'everyone'.) This formulation puts the general word 'everyone' where ordinarily we might put a name, such as 'Eve'. Predicate logic does not work this way. The second way of expressing general statements in English uses expressions such as 'Everyone is such that they love Adam.' or 'Everything is such that it loves Adam.' Predicate logic uses a formulation of this kind.

Read the symbol '(∀x)' as 'Every x is such that'. Then we transcribe 'Everyone loves Adam.' as '(∀x)Lxa'. In words, we read this as "Every x is such that x loves Adam." '(∀x)' is called a *Universal Quantifier*. In other logic books you may see it written as (x).

We are going to need not only a way of saying that **everyone** loves Adam but also a way of saying that **someone** loves Adam. Again, English does this most smoothly by putting the general word 'someone' where we might have placed a name like 'Eve'. And again logic does not imitate this style. Instead, it imitates English expressions such as 'Someone is such that he or she loves Adam.', or 'Some person is such that he or she loves Adam.', or 'Something is such that it loves Adam.' Read the symbol '(∃x)' as 'Some x is such that'. Then we transcribe 'Someone loves Adam.' as '(∃x)Lxa'. '(∃x)' is called an *Existential Quantifier*.

In one respect, '(∃x)' corresponds imperfectly to English expressions which use words such as 'some', 'there is a', and 'there are'. For example, we say 'Some cat has caught a mouse' and 'There is a cat which has caught a mouse' when we think that there is exactly one such cat. We say 'Some cats have caught a mouse' or 'There are cats which have caught a mouse' when we think that there are more than one. Predicate logic has only the one expression, '(∃x)', which does not distinguish between 'exactly one' and 'more than one'. '(∃x)' means that there is **one or more** x such that. (In chapter 9 we will learn about an extension of our logic which will enable us to make this distinction not made by '(∃x)'.)

In English, we also make a distinction by using words such as 'Every**one**' and 'every**body**' as opposed to words like 'every**thing**'. That is, English uses one word to talk about all people and another word to talk about all things which are not people. The universal quantifier, '(∀x)', does not mark this distinction. If we make no qualification, '(∀x), means all people **and** things. The same comments apply to the existential quantifier. English contrasts 'some**one**' and 'some**body**' with 'some**thing**'. But in logic, if we make no qualification, '(∃x)' means something, which can be a person or a thing. All this is very inconvenient when we want to transcribe sentences such as 'Someone loves Adam.' and 'Everybody loves Eve.' into predicate logic.

Many logicians try to deal with this difficulty by putting restrictions on the things to which the 'x' in '(∀x)' and '(∃x)' can refer. For example, in dealing with a problem which deals only with people, they say at the outset: For this problem 'x' will refer only to people. This practice is called establishing a *Universe of Discourse* or *Restricting the Domain of Discourse*. I am not going to fill in the details of this common logical practice because it really does not solve our present problem. If we resolved to talk only about people, how would we say something such as 'Every**body** likes some**thing**.'? In chapter 4 I will show you how to get the effect of restricting the domain of discourse in a more general way which will also allow us to talk at the same time about people, things, places, or whatever we like.

But until chapter 4 we will make do with the intuitive idea of restricting 'x' to refer only to people when we are transcribing sentences using expressions such as 'anybody', 'no one', and 'someone'. In other words, we will, for the time being indulge in the not quite correct practice of transcribing '(∀x)' as 'everyone', 'anybody', etc., and '(∃x)' as 'someone', 'somebody', or the like, when this is the intuitively right way to proceed, instead of the strictly correct 'everything', 'something', and similar expressions.

The letter 'x' in '(∀x)' and '(∃x)' is called a *Variable*. Variables will do an amazing amount of work for us, work very similar to that done by English pronouns, such as 'he', 'she', and 'it'. For example, watch the work 'it' does for me when I say the following: "I felt something in the closed bag. It felt cold. I pulled it out." This little discourse involves existential quantification. The discourse begins by talking about **something** without saying just which thing this something is. But then the discourse goes on to make several comments about this thing. The important point is that all the comments are about the **same** thing. This is the work that 'it' does for us. It enables us to cross-reference, making clear that we are always referring to the same thing, even though we have not been told exactly what that thing is.

A variable in logic functions in exactly the same way. For example, once we introduce the variable 'x' with the existential quantifier, '(∃x)' we can use 'x' repeatedly to refer to the same (unknown) thing. So I can say, 'Someone is blond and he or she loves Eve' with the sentence '(∃x)(Bx &Lxe)'. Note the use of parentheses here. They make clear that the quantifier '(∃x)' applies to all of the sentence 'Bx & Lxe'. Like negation, a quantifier applies to the shortest full sentence which follows it, where the shortest full following sentence may be marked with parentheses. And the 'x' in the quantifier applies to, or is linked to, all the occurrences of 'x' in this shortest full following sentence. We say that

> A quantifier *Governs* the shortest full sentence which follows it and *Binds* the variables in the sentence it governs. The latter means that the variable in the quantifier applies to all occurrences of the same variable in the shortest full following sentence.

Unlike English pronouns, variables in logic do not make cross-references between sentences.

These notions actually involve some complications in sentences which use two quantifiers, complications which we will study in chapter 3. But this rough characterization will suffice until then.

Let us look at an example with the universal quantifier, '(∀x)'. Consider the English sentences 'Anyone blond loves Eve.', 'All blonds love Eve.', 'Any blond loves Eve.', and 'All who are blond love Eve.' All these sentences say the same thing, at least so far as logic is concerned. We can express what they say more painstakingly by saying, 'Any people are such that if they are blond then they love Eve.' This formulation guides us in transcribing into logic. Let us first transcribe a part of this sentence, the conditional, which talks about some unnamed people referred to with the pronoun 'they': 'If they are blond then they love Eve.' Using the variable 'x' for the English pronoun 'they', this comes out as 'Bx ⊃ Lxe'. Now all we have to do is to say that this is true whoever "they" may be. This gives us '(∀x)(Bx ⊃ Lxe)'. Note that I have enclosed 'Bx ⊃ Lxe' in parentheses before prefixing the quantifier. This is to make clear that the quantifier applies to the whole sentence.

I have been using 'x' as a variable which appears in quantifiers and in sentences governed by quantifiers. Obviously, I would just as well have used some other letter, such as 'y' or 'z'. In fact, later on, we will need to use more than one variable at the same time with more than one quantifier. So we will take '(∀x)', '(∀y)', and '(∀z)' all to be universal quantifiers, as well as any other variable prefixed with '∀' and surrounded by parentheses if we should need still more universal quantifiers. In the same way, '(∃x)', '(∃y)', and '(∃z)' will all function as existential quantifiers, as will any similar symbol obtained by substituting some other variable for 'x', 'y', or 'z'.

To make all this work smoothly, we should clearly distinguish the letters which will serve as variables from other letters. Henceforth, I will reserve lowercase 'w', 'x', 'y', and 'z' to use as variables. I will use lowercase 'a' through 'r' as names. If one ever wanted more variables or names, one could add to these lists indefinitely by using subscripts. Thus 'a_1' and 'd_{17}' are both names, and 'x_1' and 'z_{34}' are both variables. But in practice we will never need that many variables or names.

What happened to 's', 't', 'u', and 'v'? I am going to reserve these letters to talk generally **about** names and variables. The point is this: As I have mentioned, when I want to talk generally in English about sentences in sentence logic, I use boldface capital '**X**', '**Y**', and '**Z**'. For example, when I stated the & rule I wrote, "For any sentences **X** and **Y**. . . ." The idea is that what I wrote is true no matter what sentence you might write in for '**X**' and '**Y**'. I will need to do the same thing when I state the new rules for quantifiers. I will need to say something which will be true no matter what names you might use and no matter what variables you might use. I will do this by using boldface '**s**' and '**t**' when I talk about names and boldface '**u**' and '**v**' when I talk about variables.

To summarize our conventions for notation:

> We will use lowercase letter 'a' through 'r' as names, and 'w', 'x', 'y' and 'z' as variables. We will use boldface '**s**' and '**t**' to talk generally about names and boldface '**u**' and '**v**' to talk generally about variables.

1–3. THE SENTENCES OF PREDICATE LOGIC

We now have all the pieces for saying exactly which expressions are going to count as sentences of predicate logic. First, all the sentences of sentence logic count as sentences of predicate logic. Second, we expand our stock of atomic sentences. I have already said that we will include among the atomic sentences predicates followed by the right number of names (one name for one place predicates, two names for two place predicates, and so on). We will do the same thing with variables and with variables mixed with names. So 'Bx' will count as an atomic sentence, as will 'Lxx', 'Lxy', and 'Lxa'. In general, any predicate followed by the right number of names and/or variables will count as an atomic sentence.

We get all the rest of the sentences of predicate logic by using connectives to build longer sentences from shorter sentences, starting from atomic sentences. We use all the connectives of sentence logic. And we add to these '(∀x)', '(∀y)', '(∃x)', '(∃y)', and other quantifiers, all of which count as new connectives. We use a quantifier to build a longer sentence from a shorter one in exactly the same way that we use the negation sign to build up sentences. Just put the quantifier in front of any expression which is already itself a sentence. We always understand the quantifier to apply to the shortest full sentence which follows the quantifier, as indicated by parentheses. Thus, if we start with 'Lxa', '(∀x)Lxa' counts as a sentence. We could have correctly written '(∀x)(Lxa)', though the parentheses around 'Lxa' are not needed in this case. To give another example, we can start with the atomic sentences 'Bx' and 'Lxe'. We build a compound by joining these with the conditional, '⊃', giving 'Bx ⊃ Lxe'. Finally, we apply '(∀x)' to this compound sentence. We want to be clear that '(∀x)' applies to the whole of 'Bx ⊃ Lxe', so we have to put parentheses around it before prefixing '(∀x)'. This gives '(∀x)(Bx ⊃ Lxe)'.

Here is a formal definition of sentences of predicate logic:

> All sentence letters and predicates followed by the appropriate number of names and/or variables are sentences of predicate logic. (These are the atomic sentences.) If **X** is any sentence of predicate logic and **u** is any variable, then (∀**u**)**X** (a universally quantified sentence) and (∃**u**)**X** (an existentially quantified sentence) are both sentences of predicate logic. If **X** and **Y** are both sentences of predicate logic, then any expression formed from **X** and **Y** using the connectives of sentence logic are sentences of predicate logic. Finally, only these expressions are sentences of predicate logic.

Logicians often use the words *Well Formed Formula* (Abbreviated *wff*) for any expression which this definition classifies as a predicate logic sentence.

You may have noticed something a little strange about the definition. It tells us that an expression such as '(∀x)Ba' is a predicate logic sentence. If 'A' is a sentence letter, even '(∀x)A' is going to count as a sentence! But how should we understand '(∀x)Ba' and '(∀x)A'? Since the variable 'x' of the quantifier does not occur in the rest of the sentence, it is not clear what these sentences are supposed to mean.

To have a satisfying definition of predicate logic sentence, one might want to rule out expressions such as '(∀x)Ba' and '(∀x)A'. But it will turn out that keeping these as official predicate logic sentences will do no harm, and ruling them out in the definition makes the definition messier. It is just not worth the effort to rule them out. In the next chapter we will give a more exact characterization of how to understand the quantifiers, and this characterization will tell us that "vacuous quantifiers," as in '(∀x)Ba' and '(∀x)A', have no effect at all. These sentences can be understood as the sentences 'Ba' and 'A', exactly as if the quantifiers were not there.

The definition also counts sentences such as 'By', 'Lze', and 'Bx & Lxe' as sentences, where 'x' and 'z' are variables not governed by a quantifier. Such sentences are called *Open Sentences.* Open sentences can be a problem in logic in the same way that English sentences are a problem when they contain "open" pronouns. You fail to communicate if you say, 'He has a funny nose,' without saying or otherwise indicating who "he" is.

Many logicians prefer not to count open sentences as real sentences at all. Where I use the expression 'open sentence', often logicians talk about 'open formulas' or 'propositional functions'. If you go on in your study of logic, you will quickly get used to these alternative expressions, but in an introductory course I prefer to keep the terminology as simple as possible.

Have you been wondering what the word 'syntax' means in the title of this chapter? The *Syntax* of a language is the set of rules which tell you what counts as a sentence of the language. You now know what constitutes a sentence of predicate logic, and you have a rough and ready idea of how to understand such a sentence. Our next job will be to make the interpretation of these sentences precise. We call this giving the *Semantics* for predicate logic, which will be the subject of the next chapter. But, first, you should practice what you have learned about the syntax of predicate logic to make sure that your understanding is secure.

EXERCISES

1–3. Which of the following expressions are sentences of predicate logic?

a) Ca
b) Tab
c) aTb
d) Ca ⊃ Tab
e) (∃x)~Cx

f) (∀x)(Cx ⊃ Tax)
g) (∀x)Cx & Tax(∀x)
h) ~(∀x)(Txa v Tax)
i) [(∃x)Cx v(∃x)~Cx] ≡ (∀x)(Txa & Tax)

In the following exercises, use this transcription guide:

a: Adam
e: Eve
c: Cid
Bx: x is blond
Cx: x is a cat
Lxy: x loves y
Txy: x is taller than y

Before you begin, I should point out something about transcribing between logic and pronouns in English. I used the analogy to English pronouns to help explain the idea of a variable. But that does not mean that you should always transcribe variables as pronouns or that you should always transcribe pronouns as variables. For example, you should transcribe 'If Eve is a cat, then she loves herself.' with the predicate logic sentence 'Ce ⊃ Lee'. Notice that 'she' and 'herself' are both transcribed as 'e'. That is because in this case we have been told who she and herself are. We know that they are Eve, and so we use the name for Eve, namely, 'e' to transcribe these pronouns. How should we describe 'Ca ⊃ ~Ba'? We could transcribe this as 'If Adam is a cat then Adam is not blond.' But a nicer transcription is simply 'If Adam is a cat then he is not blond.'

Now do your best with the following transcriptions.

1–4. Transcribe the following predicate logic sentences into English:

a) ~Laa
b) Laa ⊃ ~Taa
c) ~(Bc v Lce)
d) Ca ≡ (Ba v Lae)
e) (∃x)Txc
f) (∀x)Lax & (∀x)Lcx
g) (∀x)(Lax & Lcx)
h) (∃x)Txa v (∃x)Txc
i) (∃x)(Txa v Txc)
j) (∀x)(Cx ⊃ Lxe)
k) (∃x)(Cx & ~Lex)
l) ~(∀x)(Cx ⊃ Lex)
m) (∀x)[Cx ⊃ (Lcx v Lex)]
n) (∃x)[Cx & (Bx & Txc)]

1–5. Transcribe the following English sentences into sentences of predicate logic:

a) Everyone loves Eve.
b) Everyone is loved by either Cid or Adam.
c) Either everyone is loved by Adam or everyone is loved by Cid.
d) Someone is taller than both Adam and Cid.
e) Someone is taller than Adam and someone is taller than Cid.
f) Eve loves all cats.
g) All cats love Eve.
h) Eve loves some cats.
i) Eve loves no cats.
j) Anyone who loves Eve is not a cat.
k) No one who loves Eve is a cat.
l) Somebody who loves Adam loves Cid.
m) No one loves both Adam and Cid.

CHAPTER SUMMARY EXERCISES

Provide short explanations for each of the following. Check against the text to make sure that your explanations are correct, and keep your explanations in your notebook for reference and review.

a) Predicate Logic
b) Name
c) Predicate
d) One Place Predicate
e) Two Place Predicate
f) Relation
g) Variable
h) Universal Quantifier
i) Existential Quantifier
j) Universe, or Domain of Discourse
k) Govern
l) Bind
m) Open Sentence
n) Sentence of Predicate Logic
o) Well Formed Formula (wff)
p) Syntax
q) Semantics