A Modern Formal Logic Primer

Sentence Logic

Volume I

Basic Ideas and Tools 1

1-1. LOGIC AS THE SCIENCE OF ARGUMENT

Adam is happy, or so I tell you. If you don't believe me, I try to convince you with an argument: Adam just got an 'A' on his logic exam. Anyone who gets an 'A' on an exam is happy. So Adam is happy. A logician would represent such an argument in this way:

(1)	Premises	a)	Adam just got an 'A' on his logic exam.
		b)	Anyone who gets an 'A' on an exam is happy.
	Conclusion	c)	Adam is happy.

We ordinarily think of an argument as an attempt to convince someone of a conclusion by offering what a logician calls premises, that is, reasons for believing the conclusion. But in order to study arguments very generally, we will characterize them by saying:

> An *Argument* is a collection of declarative sentences one of which is called the conclusion and the rest of which are called the premises.

An argument may have just one premise, or it may have many.

By declarative sentences, I mean those, such as 'Adam is happy.' or

'Grass is green.', which we use to make statements. Declarative sentences contrast with questions, commands, and exclamations, such as 'Is Adam happy?', 'Cheer up, Adam!' and 'Boy, is Adam happy!' Throughout this text I will deal only with declarative sentences; though if you continue your study of logic you will encounter such interesting topics as the logic of questions and the logic of commands.

For an argument to have any interest, not just any premises and conclusion will do. In any argument worth its name, we must have some connection or relation between the premises and conclusion, which you can think of intuitively in this way:

> Ordinarily, the premises of an argument are supposed to support, or give us reasons, for believing the conclusion.

A good way of thinking about logic, when you are beginning to learn, is to say that logic is the study of this reason-giving connection. I like to say, more generally, that logic is the science of arguments. Logic sets out the important properties of arguments, especially the ways in which arguments can be good or bad. Along the way, logicians also study many things that are not themselves arguments or properties of arguments. These are things which we need to understand in order to understand arguments clearly or things which the study of arguments suggests as related interesting questions.

In order to see our subject matter more clearly, we need to distinguish between inductive and deductive arguments. Argument (1) is an example of a deductive argument. Compare (1) with the following:

(2) a) Adam has smiled a lot today.
b) Adam has not frowned at all today.
c) Adam has said many nice things to people today, and no unfriendly things.
———
d) Adam is happy today.

The difference between arguments (1) and (2) is this: In (1), **without fail,** if the premises are true, the conclusion will also be true. I mean this in the following sense: It is not possible for the premises to be true and the conclusion false. Of course, the premises may well be false. (I, for one, would suspect premise (b) of argument (1).) But in any possible situation in which the premises are true, the conclusion will also be true.

In argument (2) the premises relate to the conclusion in a different way. If you believe the second argument's premises, you should take yourself to have at least some fairly good reasons for believing that the conclusion is true also. But, of course, the premises of (2) could be true and the conclusion nonetheless false. For example, the premises do not rule out the possibility that Adam is merely pretending to be happy.

Logicians mark this distinction with the following terminology:

> *Valid Deductive Argument:* An argument in which, without fail, if the premises are true, the conclusion will also be true.
>
> *Good Inductive Argument:* An argument in which the premises provide good reasons for believing the conclusion. In an inductive argument, the premises make the conclusion likely, but the conclusion might be false even if the premises are true.

What do we mean by calling an argument 'deductive' or 'inductive', without the qualifiers 'valid' or 'good'? Don't let anyone tell you that these terms have rigorous definitions. Rather,

> We tend to call an argument *'Deductive'* when we claim, or suggest, or just hope that it is deductively valid. And we tend to call an argument *'Inductive'* when we want to acknowledge that it is not deductively valid but want its premises to aspire to making the conclusion likely.

In everyday life we don't use deductively valid arguments too often. Outside of certain technical studies, we intend most of our arguments as inductively good. In simple cases you understand inductive arguments clearly enough. But they can be a bear to evaluate. Even in the simple case of argument (2), if someone suggests that Adam is just faking happiness, your confidence in the argument may waver. How do you decide whether or not he is faking? The problem can become very difficult. In fact, there exists a great deal of practical wisdom about how to evaluate inductive arguments, but no one has been able to formulate an exact theory which tells us exactly when an inductive argument is really good.

In this respect, logicians understand deduction much better. Even in an introductory formal logic course, you can learn the rules which establish the deductive validity of a very wide and interesting class of arguments. And you can understand very precisely what this validity consists in and why the rules establish validity. To my mind, these facts provide the best reason for studying deductive logic: It is an interesting theory of a subject matter about which you can, in a few months, learn a great deal. Thus you will have the experience of finding out what it is like to understand a subject matter by learning a technical theory about that subject matter.

Studying formal logic also has other, more practical, attractions. Much of what you learn in this book will have direct application in mathematics, computer science, and philosophy. More generally, studying deductive logic can be an aid in clear thinking. The point is that, in order to make the nature of deductive validity very precise, we must learn a way of making certain aspects of the content of sentences very precise. For this reason, learning deductive logic can pay big dividends in improving your clarity generally in arguing, speaking, writing, and thinking.

EXERCISES

1–1. Explain in your own words what an argument is. Give an example of your own of an inductive argument and of a deductive argument. Explain why your example of an inductive argument is an inductive argument and why your example of a deductive argument is a deductive argument.

1–2. SENTENCES AND CONNECTIVES

I have said that arguments are composed of declarative sentences. Some logicians prefer to say that arguments are composed of the things we say with sentences, that is, statements or propositions. Sentences can be problematic in logic because sentences are often ambiguous. Consider this sentence:

(3) I took my brother's picture yesterday.

I could use this sentence to mean that yesterday I made a photograph of my brother. Or I could use the sentence to mean that I stole a picture that belonged to my brother. Actually, this sentence can be used to say a rather amazingly large number of different things.

Ambiguous sentences can make a problem for logic because they can be true in one way of understanding them and false in another. Because logic has to do with the truth and falsity of premises and conclusions in arguments, if it is not clear whether the component sentences are true or false, we can get into some awful messes. This is why some logicians prefer to talk about statements or propositions which can mean only one thing. In a beginning course, I prefer to talk about sentences just because they are more familiar than statements and propositions. (What are statements and propositions supposed to be, anyway?) We can deal with the problem of ambiguity of sentences by insisting that we use only unambiguous sentences, or that we specify the meaning which a possibly ambiguous sentence will have in an argument and then stick to that meaning.

Actually, in most of our work we will be concerned with certain facts about the logical form of sentences and we won't need to know exactly

what the sentences mean in detail. All we will need in order to avoid problems about ambiguity is that a given sentence be either true or false (although we usually won't know which it is) and that the sentence should not change from true to false or from false to true in the middle of a discussion. As you will see very soon, the way we will write sentences will make it extremely easy to stick to these requirements.

In fact, by restricting our attention to sentences which are either true or false, we have further clarified and extended our restriction to declarative sentences. Questions ('Is Adam happy?'), commands ('Cheer up, Adam!'), and exclamations ('Boy, is Adam happy!') are not true or false. Neither, perhaps, are some declarative sentences. Many people don't think "The woman who landed on the moon in 1969 was blond." is either true or false because no woman landed on the moon in 1969. In any case, we shall study only sentences which are definitely one or the other, true or false.

We will initially study a very simple kind of logic called *Sentence Logic*. (Logicians who work with propositions instead of sentences call it *Propositional Logic*.) The first fact on which you need to focus is that we won't be concerned with all the details of the structure of a sentence. Consider, for example, the sentence 'Adam loves Eve.' In sentence logic we won't be concerned with the fact that this sentence has a subject and a predicate, that it uses two proper names, and so on.

Indeed, the **only** fact about this sentence which is relevant to sentence logic is whether it happens to be true or false. So let's ignore all the structure of the sentence and symbolize it in the simplest way possible, say, by using the letter 'A'. (I put quotes around letters and sentences when I talk about them as opposed to using them. If this use of quotes seems strange, don't worry about it—you will easily get used to it.) In other words, for the moment, we will let the letter 'A' stand for the sentence 'Adam loves Eve.' When we do another example we will be free to use 'A' to stand for a different English sentence. But as long as we are dealing with the same example, we will use 'A' to stand for the same sentence.

Similarly, we can let other capital letters stand for other sentences. Here is a transcription guide that we might use:

Transcription Guide

A: Adam loves Eve.
B: Adam is blond.
C: Eve is clever.

'A' is standing for 'Adam loves Eve.', 'B' is standing for 'Adam is blond.', and 'C' is standing for 'Eve is clever.' In general, we will use capital letters to stand for any sentences we want to consider where we have no interest in the internal structure of the sentence. We call capital letters used in this way *Atomic Sentences*, or *Sentence Letters*. The word 'atomic' is supposed to remind you that, from the point of view of sentence logic, these are the smallest pieces we need to consider. We will always take a sentence letter (and in general any of our sentences) to be true or false (but not both true and false!) and not to change from true to false or from false to true in the middle of a discussion.

Starting with atomic sentences, sentence logic builds up more complicated sentences, or *Compound Sentences*. For example, we might want to say that Adam does **not** love Eve. We say this with the *Negation* of 'A', also called the *Denial* of 'A'. We could write this as 'not A'. Instead of 'not', though, we will just use the negation sign, '~'. That is, the negation of 'A' will be written as '~A', and will mean 'not A', that is, that 'A' is not true. The negation sign is an example of a *Connective*, that is, a symbol we use to build longer sentences from shorter parts.

We can also use the atomic sentences in our transcription guide to build up a compound sentence which says that Adam loves Eve **and** Adam is blond. We say this with the *Conjunction* of the sentence 'A' and the sentence 'B', which we write as 'A&B'. 'A' and 'B' are called *Conjuncts* or *Components* of 'A&B', and the connective '&' is called the *Sign of Conjunction*.

Finally, we can build a compound sentence from the sentence 'A' and the sentence 'B' which means that either Adam loves Eve **or** Adam is blond. We say this with the *Disjunction* of the sentence 'A' and the sentence 'B', which we write as 'AvB'. 'A' and 'B' are called *Disjuncts* or *Components* of 'AvB', and the connective 'v' is called the *Sign of Disjunction*.

You might wonder why logicians use a 'v' to mean 'or'. There is an interesting historical reason for this which is connected with saying more exactly what 'v' is supposed to mean. When I say, 'Adam loves Eve or Adam is blond.', I might actually mean two quite different things. I might mean that Adam loves Eve, or Adam is blond, but not both. Or I might mean that Adam loves Eve, or Adam is blond, or possibly both.

If you don't believe that English sentences with 'or' in them can be understood in these two very different ways, consider the following examples. If a parent says to a greedy child, 'You can have some candy or you can have some cookies,' the parent clearly means some of one, some of the other, but **not** both. When the same parent says to an adult dinner guest, 'We have plenty, would you like some more meat or some more potatoes?' clearly he or she means to be offering some of either **or** both.

Again, we have a problem with ambiguity. We had better make up our minds how we are going to understand 'or', or we will get into trouble. In principle, we could make either choice, but traditionally logicians have always opted for the second, in which 'or' is understood to mean that the first sentence is true, or the second sentence is true, or possibly both sentences are true. This is called the *Inclusive Sense* of 'or'. Latin, unlike English, was not ambiguous in this respect. In Latin, the word 'vel' very specifically meant the first or the second or possibly both. This is why logicians symbolize 'or' with 'v'. It is short for the Latin 'vel,' which means inclusive or. So when we write the disjunction 'AvB', we understand this to mean that 'A' is true, 'B' is true, or both are true.

To summarize this section:

> Sentence logic symbolizes its shortest unambiguous sentences with *Atomic Sentences*, also called *Sentence Letters*, which are written with capital letters: 'A', 'B', 'C' and so on. We can use *Connectives* to build *Compound Sentences* out of shorter sentences. In this section we have met the connectives '~' (the *Negation Sign*), '&' (the *Sign of Conjunction*), and 'v' (the *Sign of Disjunction*).

EXERCISES

1–2. Transcribe the following sentences into sentence logic, using 'G' to transcribe 'Pudding is good.' and 'F' to transcribe 'Pudding is fattening.'

a) Pudding is good and pudding is fattening.
b) Pudding is both good and fattening.
c) Pudding is either good or not fattening.
d) Pudding is not good and not fattening.

You may well have a problem with the following transcriptions, because to do some of them right you need to know something I haven't told you yet. But please take a try before continuing. Trying for a few minutes will help you to understand the discussion of the problem and its solution in the next section. And perhaps you will figure out a way of solving the problem yourself!

e) Pudding is not both good and fattening.
f) Pudding is both not good and not fattening.
g) Pudding is not either good or fattening.
h) Pudding is either not good or not fattening.
i) Pudding is neither good nor fattening.

1–3. TRUTH TABLES AND THE MEANING OF '~', '&', AND 'v'

We have said that '~A' means not A, 'A&B' means A and B, and 'AvB' means A or B in the inclusive sense. This should give you a pretty good idea of what the connectives '~', '&', and 'v' mean. But logicians need to be as exact as possible. So we need to specify how we should understand the connectives even more exactly. Moreover, the method which we will use to do this will prove very useful for all sorts of other things.

To get the idea, we start with the very easy case of the negation sign, '~'. The sentence 'A' is either true or it is false. If 'A' is true, then '~A' is false. If 'A' is false, then '~A' is true. And that is everything you need to know about the meaning of '~'. We can say this more concisely with a table, called a *Truth Table:*

Truth table definition of '~'

	A	~A
case 1	t	f
case 2	f	t

The column under 'A' lists all the possible cases involving the truth and falsity of 'A'. We do this by describing the cases in terms of what we call *Truth Values*. The case in which A is true is described by saying that A has the truth value t. The case in which A is false is described by saying that A has the truth value f. Because A can only be true or false, we have only these two cases. We explain how to understand '~' by saying what the truth value of '~A' is in each case. In case 1, '~A' has the truth value f; that is, it is false. In case 2, '~A' has the truth value t; that is, it is true. Although what we have done seems trivial in this simple case, you will see very soon that truth tables are extremely useful.

Let us see how to use truth tables to explain '&'. A conjunction has two atomic sentences, so we have four cases to consider:

	A	B	A&B
case 1	t	t	
case 2	t	f	
case 3	f	t	
case 4	f	f	

When 'A' is true, 'B' can be true or false. When 'A' is false, again 'B' can be true or false. The above truth table gives all possible combinations of truth values which 'A' and 'B' can have together.

We now specify how '&' should be understood by specifying the truth value for each case for the compound 'A&B':

Truth table definition of '&'

	A	B	A&B
case 1	t	t	t
case 2	t	f	f
case 3	f	t	f
case 4	f	f	f

In other words, 'A&B' is true when the conjuncts 'A' and 'B' are both true. 'A&B' is false in all other cases, that is, when one or both of the conjuncts are false.

A word about the order in which I have listed the cases. If you are curious, you might try to guess the recipe I used to order the cases. (If you try, also look at the more complicated example in section 1–5.) But I won't pause to explain, because all that is important about the order is that we don't leave any cases out and all of us list them in the same order, so that we can easily compare answers. So just list the cases as I do.

We follow the same method in specifying how to understand 'v'. The disjunction 'AvB' is true when either or both of the disjuncts 'A' and 'B' are true. 'AvB' is false only when 'A' and 'B' are both false:

Truth table definition of 'v'

	A	B	AvB
case 1	t	t	t
case 2	t	f	t
case 3	f	t	t
case 4	f	f	f

We have defined the connectives '~', '&', and 'v' using truth tables for the special case of sentence letters 'A' and 'B'. But obviously nothing will change if we use some other pair of sentences, such as 'H' and 'D'.

This section has focused on the truth table definitions of '~', '&' and 'v'. But along the way I have introduced two auxiliary notions about which you need to be very clear. First, by a *Truth Value Assignment of Truth Values to Sentence Letters,* I mean, roughly, a line of a truth table, and a *Truth Table* is a list of all the possible truth values assignments for the sentence letters in a sentence:

> An *Assignment of Truth Values* to a collection of atomic sentence letters is a specification, for each of the sentence letters, whether the letter is (for this assignment) to be taken as true or as false. The word *Case* will also be used for 'assignment of truth values'.

> A *Truth Table for a Sentence* is a specification of all possible truth values assignments to the sentence letters which occur in the sentence, and a specification of the truth value of the sentence for each of these assignments.

1–4. TRUTH FUNCTIONS

I want to point out one more thing about the way we have defined the connectives '~', '&', and 'v'. Let us start with '~'. What do you have to know in order to determine whether '~A' is true or false? You don't have to know what sentence 'A' actually stands for. You don't have to know whether 'A' is supposed to mean that Adam loves Eve, or that pudding is fattening, or anything like that. To know whether '~A' is true or false, **all** you have to know is whether 'A' itself is true or false. This is because if you know the truth value of 'A', you can get the truth value of '~A' by just looking it up in the truth table definition of '~'.

The same thing goes for '&' and 'v'. To know whether 'A&B' is true or false, you don't have to know exactly what sentences 'A' and 'B' are supposed to be. All you need to know is the truth value of 'A' and the truth value of 'B'. This is because, with these truth values, you can look up the truth value of 'A&B' with the truth table definition of '&'. Likewise, with truth values for 'A' and for 'B', you can look up the truth value for 'AvB'.

Logicians have a special word for these simple facts about '~', '&' and 'v'. We say that these connectives are *Truth Functional.* In other words (to use '&' as an example), the truth value of the compound sentence 'A&B' is a function of the truth values of the components 'A' and 'B'. In other words, if you put in truth values for 'A' and for 'B' as input, the truth table definition of '&' gives you, as an output, the truth value for the compound 'A&B'. In this way 'A&B' is a function in the same kind of way that 'x + y' is a numerical function. If you put in specific numbers for 'x' and 'y', say, 5 and 7, you get a definite value for 'x + y', namely, 12.

'A&B' is just like that, except instead of number values 1, 2, 3, . . . which can be assigned to 'x' and to 'y', we have just two truth values, t and f, which can be assigned to 'A' and to 'B'. And instead of addition, we have some other way of combining the assigned values, a way which we gave in the truth table definition of '&'. Suppose, for example, that I give you the truth values t for 'A' and f for 'B'. What, then, is the resulting truth value for 'A&B'? Referring to the truth table definition of 'A&B', you can read off the truth value f for 'A&B'. The truth tables for '~' and for 'v' give other ways of combining truth values of components to get truth values for the compound. That is,'~' and 'v' are different truth functions.

Let's pull together these ideas about truth functions:

> A *Truth Function* is a rule which, when you give it input truth values, gives you a definite output truth value. A *Truth Functional Connective* is a connective defined by a truth function. A *Truth Functional Compound* is a compound sentence formed with truth functional connectives.

EXERCISES

1–3. Try to explain what it would be for a declarative compound sentence in English **not** to be truth functional. Give an example of a declarative compound sentence in English that is not truth functional. (There are lots of them! You may find this exercise hard. Please try it, but don't get alarmed if you have trouble.)

1–5. COMPOUNDING COMPOUND SENTENCES

We have seen how to apply the connectives '~', '&', and 'v' to atomic sentences such as 'A' and 'B' to get compound sentences such as '~A', 'A&B', and 'AvB'. But could we now do this over again? That is, could we apply the connectives not just to atomic sentences 'A', 'B', 'C', etc., but also to the compound sentences '~A', 'A&B', and 'AvB'? Yes, of course. For example, we can form the conjunction of '~A' with 'B', giving us '~A&B'. Using our current transcription guide, this transcribes into 'Adam does not love Eve and Adam is blond.'

As another example, we could start with the conjunction 'A&B' and take this sentence's negation. But now we have a problem. (This is the problem you encountered in trying to work exercise 1–2, e–i.) If we try to write the negation of 'A&B' by putting a '~' in front of 'A&B', we get the sentence we had before. But the two sentences should not be the same! This might be a little confusing at first. Here is the problem: We are considering two ways of building up a complex sentence from shorter parts, resulting in two different complex sentences. In the first way, we take the negation of 'A', that is, '~A', and conjoin this with 'B'. In the second way, we **first** conjoin 'A' and 'B' and **then** negate the whole. In English, the sentence 'It is **not** the case **both** that Adam loves Eve and Adam is blond.' is very different from the sentence 'Adam does **not** love Eve, and Adam **is** blond.' (Can you prove this by giving circumstances in which one of these compound sentences is true and the other one is false?)

In order to solve this problem, we need some device in logic which does the work that 'both' does in English. (If you are not sure you yet understand what the problem is, read the solution I am about to give and then reread the last paragraph.) What we need to do is to make clear the order in which the connectives are applied. It makes a difference whether we **first** make a negation and **then** form a conjunction, or whether we **first** form the conjunction and **then** make a negation. We will indicate the order of operations by using parentheses, much as one does in algebra. Whenever we form a compound sentence we will surround it by parentheses. Then you will know that the connective inside the parentheses applies before the one outside the parentheses. Thus, when we form the negation of 'A', we write the final result as '(~A)'. We now take '(~A)' and conjoin it with 'B', surrounding the final result with parentheses:

(4) [(~A)&B]

This says, take the sentence '(~A)' and conjoin it with 'B'. To indicate that the final result is a complete sentence (in case we will use it in some still larger compound), we surround the final result in parentheses also. Note how I have used a second style for the second pair of parentheses—square brackets—to make things easier to read.

Contrast (4) with

(5) [~(A&B)]

which means that one is to conjoin 'A' with 'B' and then take the negation of the whole.

In the same kind of way we can compound disjunctions with conjunctions and conjunctions with disjunctions. For example, consider

(6) [(A&B)vC]

(7) [(A&(BvC))]

Sentence (6) says that we are first to form the conjunctions of 'A' with 'B' and then form the disjunction with 'C'. (7), on the other hand, says that we are first to form the **disjunction** of 'B' with 'C' and then conjoin the whole with 'A'. These are very different sentences. Transcribed into English, they are 'Adam both loves Eve and is blond, or Eve is clever.' and 'Adam loves Eve, and either Adam is blond or Eve is clever.'

We can show more clearly that (6) and (7) are different sentences by writing out truth tables for them. We now have three atomic sentences, 'A', 'B', and 'C'. Each can be true or false, whatever the others are, so that we now have eight possible cases. For each case we work out the truth value of a larger compound from the truth value of the parts, using the truth value of the intermediate compound when figuring the truth value of a compound of a compound:

	a	b	c	d	e	g	h
	A	B	C	(A&B)	(BvC)	[(A&B)vC]	[A&(BvC)]
case 1	t	t	t	t	t	t	t
case 2	t	t	f	t	t	t	t
case 3	t	f	t	f	t	t	t
case 4	t	f	f	f	f	f	f
case 5	f	t	t	f	t	t	f
case 6	f	t	f	f	t	f	f
case 7	f	f	t	f	t	t	f
case 8	f	f	f	f	f	f	f

Let's go over how we got this truth table. Columns a, b, and c simply give all possible truth value assignments to the three sentence letters 'A', 'B', and 'C'. As before, in principle, the order of the cases does not matter. But to make it easy to compare answers, you should always list the eight possible cases for three letters in the order I have just used. Then, for each case, we need to calculate the truth value of the compounds in columns d through h from the truth values given in columns a, b, and c.

Let us see how this works for case 5, for example. We first need to determine the truth value to put in column d, for '(A&B)' from the truth values given for this case. In case 5 'A' is false and 'B' is true. From the truth table definition of '&', we know that a conjunction (here, 'A&B') is false when the first conjunct (here, 'A') is false and the second conjunct (here, 'B') is true. So we write an 'f' for case 5 in column d. Column e is the disjunction of 'B' with 'C'. In case 5 'B' is true and 'C' is true. When we disjoin something true with something true, we get a true sentence. So we write the letter 't', standing for the truth value t, in column e for case 5.

Moving on to column g, we are looking at the disjunction of '(A&B)' with 'C'. We have already calculated the truth value of '(A&B)' for case 5—that was column d—and the truth value of 'C' for case 5 is given in column c. Reading off columns c and d, we see that '(A&B)' is false and 'C' is true in case 5. The sentence of column g, '[(A&B)vC]', is the disjunction of these two components and we know that the disjunction of something false with something true is, again, true. So we put a 't' in column g for case 5. Following the same procedure for column h, we see that for case 5 we have a conjunction of something false with something true, which gives the truth value f. So we write 'f' for case 5 in column h.

Go through all eight cases and check that you understand how to determine the truth values for columns d through h on the basis of what you are given in columns a, b, and c.

Now, back to the point that got us started on this example. I wanted to prove that the sentences '[(A&B)vC]' and '[A&(BvC)]' are importantly different. Note that in cases 5 and 7 they have different truth values. That is, there are two assignments of truth values to the components for which one of these sentences is true and the other is false. So we had better not confuse these two sentences. You see, we really do need the parentheses to distinguish between them.

Actually, we don't need all the parentheses I have been using. We can make two conventions which will eliminate the need for some of the parentheses without any danger of confusing different sentences. First, we can eliminate the outermost parentheses, as long as we put them back in if we decide to use a sentence as a component in a larger sentence. For example, we can write 'A&B' instead of '(A&B)' as long as we put the parentheses back around 'A&B' before taking the negation of the whole to form '~(A&B)'. Second, we can agree to understand '~' always to apply to the shortest full sentence which follows it. This eliminates the need to surround a negated sentence with parentheses before using it in a larger sentence. For example, we will write '~A&B' instead of '(~A)&B'. We know that '~A&B' means '(~A)&B' and not '~(A&B)' because the '~' in

'~A&B' applies to the **shortest** full sentence which follows it, which is 'A' and not 'A&B'.

This section still needs to clarify one more aspect of dealing with compound sentences. Suppose that, before you saw the last truth table, I had handed you the sentence '(A&B)vC' and asked you to figure out its truth value in each line of a truth table. How would you know what parts to look at? Here's the way to think about this problem. For some line of a truth table (think of line 5, for example), you want to know the truth value of '(A&B)vC'. You could do this if you knew the truth values of 'A&B' and of 'C'. With their truth values you could apply the truth table definition of 'v' to get the truth value of '(A&B)vC'. This is because '(A&B)vC' just **is** the disjunction of 'A&B' with 'C'. Thus you know that '(A&B)vC' is true if at least one of its disjuncts, that is, either 'A&B' or 'C', is true; and '(A&B)vC' is false only if both its disjuncts, 'A&B' and 'C', are false.

And how are you supposed to know the truth values of 'A&B' and of 'C'? Since you are figuring out truth values of sentences in the line of a truth table, all you need do to figure out the truth value of 'C' on that line is to look it up under the 'C' column. Thus, if we are working line 5, we look under the 'C' column for line 5 and read that in this case 'C' has the truth value t. Figuring out the truth value for 'A&B' for this line is almost as easy. 'A&B' is, by the truth table definition of conjunction, true just in case both conjuncts (here, 'A' and 'B') are true. In line 5 'A' is false and 'B' is true. So for this line, 'A&B' is false.

Now that we finally have the truth values for the parts of '(A&B)vC', that is, for 'A&B' and for 'C', we can plug these truth values into the truth table definition for v and get the truth value t for '(A&B)vC'.

Now suppose that you have to do the same thing for a more complicated sentence, say

(8) ~{[Av~C]&[Bv(~A&C)]}

Don't panic. The principle is the same as for the last, simpler example. You can determine the truth value of the whole if you know the truth value of the parts. And you can determine the truth value of the parts if you can determine the truth value of **their** parts. You continue this way until you get down to atomic sentence letters. The truth value of the atomic sentence letters will be given to you by the line of the truth table. With them you can start working your way back up.

You can get a better grip on this process with the idea of the *Main Connective* of a sentence. Look at sentence (8) and ask yourself, "What is the last step I must take in building this sentence up from its parts?" In the case of (8) the last step consists in taking the sentence '[Av~C]&[Bv(~A&C)]' and applying '~' to it. Thus (8) is a negation, '~' is the main connective of (8), and '[Av~C]&[Bv(~A&C)]' is the component used in forming (8).

What, in turn, is the main connective of '[Av~C]&[Bv(~A&C)]'? Again, what is the last step you must take in building this sentence up from its parts? In this case you must take 'Av~C' and conjoin it with 'Bv(~A&C)'. Thus this sentence is a conjunction, '&' is its main connective, and its components are the two conjuncts 'Av~C' and 'Bv(~A&C)'. In like manner, 'Bv(~A&C)' is a disjunction, with 'v' its main connective, and its components are the disjuncts 'B' and '~A&C'. To summarize,

> The *Main Connective* in a compound sentence is the connective which was used last in building up the sentence from its component or components.

Now, when you need to evaluate the truth value of a complex sentence, given truth values for the atomic sentence letters, you know how to proceed. Analyze the complete sentence into its components by identifying main connectives. Write out the components, in order of increasing complexity, so that you can see plainly how the larger sentences are built up from the parts.

In the case of (8), we would lay out the parts like this:

A, B, C, ~A, ~C, Av~C, ~A&C, Bv(~A&C), [Av~C]&[Bv(~A&C)], ~[Av~C]&[Bv(~A&C)]}

You will be given the truth values of the atomic sentence letters, either by me in the problem which I set for you or simply by the line of the truth table which you are working. Starting with the truth values of the atomic sentence letters, apply the truth table definitions of the connectives to evaluate the truth values of the successively larger parts.

EXERCISES

1–4. For each of the following sentences, state whether its main connective is '~', '&', or 'v' and list each sentence's components. Then do the same for the components you have listed until you get down to atomic sentence letters. So you can see how you should present your answers, I have done the first problem for you.

	Sentence	Main Connective	Components
a)	~(Av~B)	~	Av~B
	Av~B	v	A, ~B
	~B	~	B

a) ~(Av~B)
b) (D&~G)v(G&D)
c) [(Dv~~B)&(DvB)]&(DvB)
d) L&{Mv[~N&(Mv~L)]}

1–6. RULES OF FORMATION AND RULES OF VALUATION

We can summarize many important points discussed so far by giving explicit rules which tell us what counts as a sentence of sentence logic and how to determine the truth values of compound sentences if we are given the truth values of the components:

Formation Rules

i) Every capital letter 'A', 'B', 'C' . . . is a sentence of sentence logic. Such a sentence is called an *Atomic Sentence* or a *Sentence Letter.*

ii) If **X** is a sentence of sentence logic, so is (~**X**), that is, the sentence formed by taking **X**, writing a '~' in front of it, and surrounding the whole by parentheses. Such a sentence is called a *Negated Sentence.*

iii) If **X** and **Y** are sentences of sentence logic, so is (**X**&**Y**), that is, the sentence formed by writing **X**, followed by '&', followed by **Y**, and surrounding the whole with parentheses. Such a sentence is called a *Conjunction,* and **X** and **Y** are called its *Conjuncts.*

iv) If **X** and **Y** are sentences of sentence logic, so is (**X**v**Y**), that is, the sentence formed by writing **X**, followed by 'v', followed by **Y**, and surrounding the whole with parentheses. Such a sentence is called a *Disjunction,* and **X** and **Y** are called its *Disjuncts.*

v) Until further notice, only expressions formed by using rules i) through iv) are sentences of sentence logic.

If you wonder why I say "until further notice," I want you to digest the present and some new background material before I introduce two new connectives, corresponding to the expressions "If . . . then" and "if and only if." When I introduce these new connectives, the formation rules will need to be extended accordingly.

As I explained earlier, we agree to cheat on these strict rules in two ways (and in these two ways only!). We omit the outermost parentheses, and we omit parentheses around a negated sentence even when it is not the outermost sentence, because we agree to understand '~' always to apply to the shortest full sentence which follows it.

I should also clarify something about formation rule i). In principle, sentence logic can use as many atomic sentences as you like. It is not limited to the 26 letters of the alphabet. If we run out of letters, we can always invent new ones, for example, by using subscripts, as in 'A_1' and 'C_{37}'. In practice, of course, we will never need to do this.

Rules of Valuation

i) The truth value of a negated sentence is t if the component (the sentence which has been negated) is f. The truth value of a negated sentence is f if the truth value of the component is t.

ii) The truth value of a conjunction is t if both conjuncts have truth value t. Otherwise, the truth value of the conjunction is f.

iii) The truth value of a disjunction is t if either or both of the disjuncts have truth value t. Otherwise, the truth value of the disjunction is f.

Note that these rules apply to any compound sentence. However, they only apply if somehow we have been given a truth value assignment to the atomic sentence letters. That is, **if** we have been given truth values for the ultimate constituent atomic sentence letters, then, using the rules of valuation, we can always calculate the truth value of a compound sentence, no matter how complex. Once again, this is what we mean when we say that the connectives are truth functional.

How does one determine the truth value of atomic sentences? That's not a job for logicians. If we really want to know, we will have to find out the truth value of atomic sentences from someone else. For example, we'll have to consult the physicists to find out the truth value of "light always travels at the same speed." As logicians, we only say what to do with truth values of atomic constituents once they are given to us. And when we do truth tables, we don't have to worry about the actual truth values of the atomic sentence letters. In truth tables, like those in the following exercises, we consider **all possible combinations** of truth values which the sentence letters could have.

The truth table definitions of the connectives give a graphic summary of these rules of valuation. I'm going to restate those truth table definitions here because, if truth be told, I didn't state them quite right. I gave them only for **sentence letters,** 'A' and 'B'. I did this because, at that point in the exposition, you had not yet heard about long compound sentences, and I didn't want to muddy the waters by introducing too many new things at once. But now that you are used to the idea of compound sentences, I can state the truth table definitions of the connectives with complete generality.

Suppose that **X** and **Y** are any two sentences. They might be atomic sentence letters, or they might themselves be very complex compound sentences. Then we specify that:

		X	~X
Truth table	case 1	t	f
definition of '~'	case 2	f	t

Truth table definition of '&'

	X	Y	X&Y
case 1	t	t	t
case 2	t	f	f
case 3	f	t	f
case 4	f	f	f

Truth table definition of 'v'

	X	Y	XvY
case 1	t	t	t
case 2	t	f	t
case 3	f	t	t
case 4	f	f	f

The difference between my earlier, restricted truth table definitions and these new general definitions might seem a bit nitpicky. But the difference is important. You probably understood the intended generality of my first statement of the truth table definitions. However, a computer, for example, would have been totally confused. Logicians strive, among other things, to give very exact statements of everything. They enjoy exactness for its own sake. But exactness has practical value too, for example, when one needs to write a program that a computer can understand.

This section has also illustrated another thing worth pointing out. When I talked about sentences generally, that is, when I wanted to say something about **any** sentences, **X** and **Y**, I used boldface capital letters from the end of the alphabet. I'm going to be doing this throughout the text. But rather than dwell on the point now, you will probably best learn how this usage works by reading on and seeing it illustrated in practice.

EXERCISES

1–5. Which of the following expressions are sentences of sentence logic and which are not?

a) A&~B
b) A~&B
c) Gv(~B&~H)
d) A&(C&~(DvH))
e) (A&B)v(C&D)
f) (AvB)&CvD

1–6. Construct a complete truth table for each of the following sentences. The first one is done for you:

A	B	~B	~BvA
t	t	f	t
t	f	t	t
f	t	f	f
f	f	t	t

a) ~BvA
b) ~(BvA)
c) (QvT)&(~Qv~T)
d) (D&~G)v(G&D)
e) Av(~BvC)
f) Kv[~P&(~PvM)]
g) [(Dv~~B)&(Dv~B)]&(DvB)
h) L&{Mv[~N&(~Mv~L)]}

1–7. Philosopher's problem: Why do I use quotation marks around sentences, writing things like

'B'

and

'~(Cv~A)'

but no quotation marks about boldface capital letters, writing

X, **Y**, **XvY**, etc.

when I want to talk about sentences generally?

CHAPTER SUMMARY EXERCISE

The following list gives you the important terms which have been introduced in this chapter. Make sure you understand all of them by writing a short explanation of each. Please refer back to the text to make sure, in each case, that you have correctly explained the term. Keep your explanation of these terms in your notebook for reference and review later on in the course.

a) Argument
b) Valid Deductive Argument

c) Good Inductive Argument
d) Deductive Argument
e) Inductive Argument
f) Atomic Sentence (also called 'Sentence Letter' or 'Atomic Sentence Letter')
g) Compound Sentence
h) Connective
i) Component
j) ~ (called the 'Negation Sign' or 'Sign of Denial')
k) Negation
l) & (called the 'Sign of Conjunction')
m) Conjunction
n) Conjunct
o) v (called the 'Sign of Disjunction')
p) Disjunction
q) Disjunct
r) Inclusive Or
s) Exclusive Or
t) Truth Value
u) Truth Table
v) Truth Table Definition
w) Assignment of Truth Values
x) Case
y) Truth Function
z) Truth Functional Connective
aa) Truth Functional Compound
bb) Main Connective