

# Truth Trees for Sentence Logic

## Applications

### 9

#### 9-1. APPLICATION OF THE RULES TO COMPLEX SENTENCES

This is going to be a short chapter. You really have all the facts about truth trees. It only remains to see how to apply these facts in some new ways.

In the last chapter I was careful to give you only problems in which the sentences were very simple. But now that you have the hang of the rules, let's see how to apply them in testing the validity of an argument like this:

$(\bar{A}\&B)\vee(\sim A\&C)$	√1	$(A\&B)\vee(\sim A\&C)$	P
$B\vee C$	√2	$\sim(B\vee C)$	$\sim C$
	3	$\sim B$	2, $\sim\vee$
	4	$\sim C$	2, $\sim\vee$
	√5	A&B	1, $\vee$
	6	A	5, &
	7	B	5, &
	x		
		$\sim A\&C$	
		$\sim A$	
		C	
		x	

Valid

-Following the suggestion of working first on nonbranching lines, I began with line 2. But what, then, should I do with line 1? Line 1 is a disjunction of conjunctions. Which rule applies? And how? Keep in mind

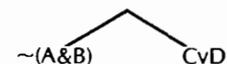
what the rules are supposed to do. Our objective is to make line 1 true, in all the minimally sufficient ways that this can be done. Because line 1 is a disjunction, we must do this by making each of the disjuncts true along a separate leg of a branch. That is, we must make the full subsentence 'A&B' true along one branch and the full subsentence ' $\sim A\&C$ ' true along a second branch. The subsentences 'A&B' and ' $\sim A\&C$ ' are themselves compound sentences to which we must apply the rule for conjunction, which I have done in lines 6 and 7.

How can you tell which rule applies to line 1? Ask yourself: What do I have to do to the **whole** sentence appearing on line 1 which will guarantee that the whole sentence is true? The answer will direct you to the components which must be written at points farther down on the tree.

To see more clearly how this works, let us look at some more examples. If the sentence is

$$(A\&B)\supset(C\vee D)$$

I say to myself: This sentence is a conditional, the antecedent of which is the conjunction 'A&B' and the consequent of which is the disjunction 'C∨D'. A conditional can be made true by making the antecedent false. And it can alternatively be made true by making the consequent true. So at the bottom of every open path on which '(A&B)⊃(C∨D)' appears, I must write the branch



What should I do with

$$\sim[(B\vee C)\supset A]?$$

This sentence is a negated conditional. To make it true I must simultaneously make the conditional's antecedent true and its consequent false. So at the bottom of every open path on which it appears, I must write the stack

$$B\vee C$$

$$\sim A$$

We will look at some nastier examples in a moment. But first let's discuss an explicit prescription for applying the rules to sentences no matter how complex. Suppose you are faced with a very complex sentence and you are not sure which rule applies. Ask yourself the following question: In building this sentence up from smaller parts, what was the last step?

What was the very last connective that got used in putting together this compound sentence from its components?

I hope that by this time you are recognizing the idea of a sentence's main connective from chapter 1. (If you did the chapters on natural deduction, you may find this section repetitious. Please bear with those who are learning the tree method without having learned natural deduction first.)

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

To determine which rule applies to a sentence, first determine the sentence's main connective. If the main connective is not '~' all you have to do is to apply the rule for the main connective. If the main connective is '~', determine the main connective of the negated sentence. Then apply the corresponding rule  $\sim\vee$ ,  $\sim\&$ ,  $\sim\supset$ ,  $\sim\equiv$ , or  $\sim\sim$ .

Let us see how the rules apply to a few more examples. Consider

$$(A\supset B)\supset[(C\vee A)\supset B]$$

What is the main connective? It is ' $\supset$ '. But which occurrence of ' $\supset$ '? It's the second occurrence. The parentheses tell you that the very last step in building up this sentence is to take ' $A\supset B$ ' and ' $(C\vee A)\supset B$ ' and to make the first the antecedent and the second the consequent of a conditional.

Here is another example:

$$\sim\{[(A\equiv\sim B)\equiv C]\equiv[C\supset(\sim A\equiv B)]\}$$

This is a negated biconditional, and the occurrence of ' $\equiv$ ' to which you have to apply the rule is the third. In building the sentence up from its parts, the very last thing that was done was to apply the outermost negation sign. The step before that was to form a biconditional from the components ' $(A\equiv\sim B)\equiv C$ ' and ' $C\supset(\sim A\equiv B)$ '. So the rule for negated biconditional applies, using ' $(A\equiv\sim B)\equiv C$ ' and ' $C\supset(\sim A\equiv B)$ ' as the components. At the bottom of every open branch, we write

$$\begin{array}{cc} & \wedge \\ (A\equiv\sim B)\equiv C & \sim[(A\equiv\sim B)\equiv C] \\ \sim[C\supset(\sim A\equiv B)] & C\supset(\sim A\equiv B) \end{array}$$

Before turning you loose on some exercises, I should mention a small side point. When you worked problem 8-4g, one of the open branches displayed ' $\sim H$ ' and ' $K$ ' but neither ' $S$ ' nor ' $\sim S$ '. So what counterexamples

does this branch represent? What happened in this case is that making ' $H$ ' false and ' $K$ ' true is already enough to make everything on the branch true. If ' $H$ ' is false and ' $K$ ' is true, the initial sentences are true whether ' $S$ ' is true or false. But, strictly speaking, an assignment of truth values to sentence letters for a sentence must assign a truth value to each sentence letter in the sentence. So, strictly speaking, a counterexample for this problem must specify a truth value for ' $S$ '. Thus we really should say that the assignment of truth values which you read off the open branch, ' $H$ ' false and ' $K$ ' true, is an abbreviation for the pair of counterexamples, ' $\sim H\&K\&S$ ' and ' $\sim H\&K\&\sim S$ '.

However, having said this, we will record counterexamples by reading off the truth values for the sentence letters and negated sentence letters which occur on an open branch. If some sentence letters have been left out, we know that our list is an abbreviation for all the truth value assignments which result by arbitrarily assigning truth values for the neglected sentence letters.

#### EXERCISES

9-1. Determine the main connective in the following sentences:

- $A\&[B\vee(C\supset D)]$
- $\sim\{[(H\vee\sim K)\equiv F]\supset(\sim R\vee\sim F)\}$
- $\sim\{[(I\vee\sim P)\equiv M]\supset(\sim I\vee\sim G)\}$
- $\{[F\supset(B\vee\sim N)]\supset(\sim J\supset N)\}\supset\{N\vee[F\supset\sim(B\vee J)]\}$

9-2. Test the following arguments for validity. Show your trees, showing which paths are closed. Say whether the argument is valid or invalid, and if invalid give the counterexamples provided by the finished tree.

Before beginning these problems, you should review the practical guides at the end of chapter 8. Also, try to stay clear of the following pitfalls that often catch students: A tree is not completed until either all branches have closed or until all sentences have been checked. Sometimes you can see before completing a tree that there will surely be at least one counterexample. (This can happen, for example, when completing the left branch produced by an original sentence, before the right branch is complete.) But, both for safety and to make sure you get plenty of practice, please don't quit on a tree until all compound sentences have been checked.

Sometimes students try to take shortcuts, writing down the results

of working on two rules at once. Often this produces mistakes, and makes it terribly hard for anyone to correct your papers. Finally, pay constant attention to the main connective. Only by correctly identifying the main connective in a compound sentence will you correctly apply the rules.

- |  |  |   |   |
|--|--|---|---|
| a) $\frac{B}{(B \vee C) \& (B \vee D)}$  | b) $\frac{D \vee F \quad K \supset \sim F}{\sim F \supset (D \vee K)}$   | c) $\frac{N \supset (D \supset P) \quad N \supset D}{N \supset P}$  | d) $\frac{(H \& P) \vee (S \& \sim J) \quad J \supset \sim (H \& D)}{(J \supset \sim D)}$ |
| e) $\frac{(R \vee L) \supset (G \vee A) \quad R \supset \sim G \quad \sim (B \& \sim R)}{\sim B \vee A}$                                     | f) $\frac{(\sim I \& \sim D) \vee \sim D \quad \sim [(I \& \sim J) \vee (D \& \sim J)]}{\sim B \vee \sim A}$                       | g) $\frac{(G \supset B) \& (\sim G \supset N) \quad B \vee \sim N}{N \vee \sim B}$  |   |
| h) $\frac{(D \supset H) \supset P \quad D \supset \sim (F \vee G) \quad F \vee H}{D \supset P}$  | i) $\frac{I \supset (J \supset K) \quad (I \supset) \supset K}{I \supset K}$   | j) $\frac{F \& [P \vee \sim (D \supset F)] \quad D \vee \sim (F \vee K)}{K \supset D}$                                      |   |
| k) $\frac{(T \& G) \vee (G \& \sim M) \quad T \& \sim M}{T \& \sim M}$   | l) $\frac{(H \equiv \sim Q) \equiv (H \equiv \sim M) \quad H \supset (Q \vee \sim (Q \& M))}{H \supset (Q \vee \sim (Q \& M))}$    | m) $\frac{Q \equiv M}{(H \equiv \sim Q) \equiv (H \equiv \sim M)}$  |   |
| n) $\frac{H \supset (D \supset (B \vee P)) \quad D \vee P}{H \supset P}$   | o) $\frac{\sim [\sim K \& \sim (\sim A \& \sim B)] \quad (A \supset K) \& (K \supset B)}{A \supset B}$                             | p) $\frac{(G \vee A) \supset (H \supset B) \quad [H \supset (H \& B)] \supset K}{G \supset K}$                              |   |
| q) $\frac{D \vee (M \supset J) \quad [M \supset (M \& J)] \supset (P \vee K) \quad (P \supset D) \& (K \supset F)}{D \vee F}$                | r) $\frac{[(F \& \sim B) \vee Q] \supset [A \& (S \vee T)] \quad F \& \sim (S \& A) \quad S \vee A}{B \supset (A \supset S)}$      |   |   |
| s) $\frac{F \supset (K \vee B) \quad (\sim F \vee G) \& (\sim G \vee \sim K)}{F \supset B}$  | t) $\frac{F \equiv [\sim J \vee (C \& T)] \quad A \vee (C \equiv O) \quad \sim [F \supset (J \supset \sim O)]}{F \equiv (T \& A)}$ | u) $\frac{Q \equiv \sim (A \& F) \quad \sim (M \vee A) \supset \sim H \quad \sim (Q \& A) \vee F}{Q \supset (H \supset M)}$ |   |
| v) $\frac{(I \& \sim T) \supset P \quad \sim A \supset \sim T \quad \sim T \vee C \quad C \supset D}{\sim P \supset [(I \supset (D \& A))]}$ |  |   |   |

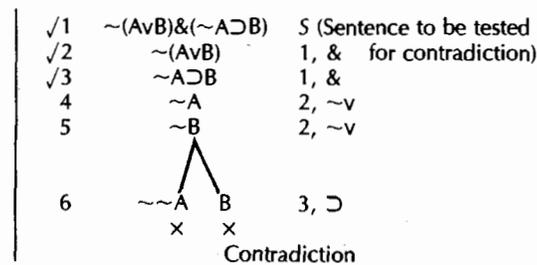
9-2. OTHER USES FOR TRUTH TREES

So far we have used truth trees exclusively to test the validity of arguments. But now that we understand how truth trees work we can easily apply them to do all sorts of other jobs. For example, suppose I hand you the sentence ' $\sim (A \vee B) \& (\sim A \supset B)$ ' and ask you whether it is a contradiction. You could always work this problem with truth tables. If in all lines of the truth table the sentence is false, it is, by definition, a contradiction. If the sentence is true in one or more cases it is not a contradiction. But a truth tree will get the job done faster. If we make a sentence the initial line on a truth tree, we know the truth tree will find us a case in which the sentence is true, if there is such a case. If there is no such case, all paths on the tree will close. Can you see how the truth tree will tell us whether the sentence is a contradiction?

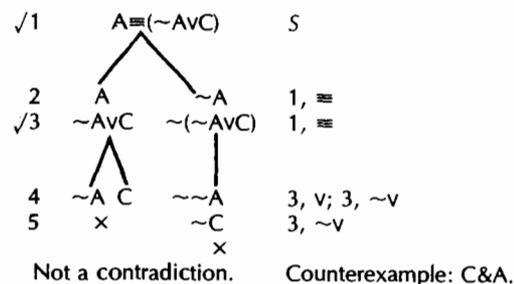
To explain this more thoroughly, we can again use the idea of a counterexample. A contradiction is false in all cases. So a case in which a sentence is true constitutes a *Counterexample* to its being a contradiction. The truth tree method applies immediately to look for counterexamples to a sentence being a contradiction. We make the sentence to be tested the first line of a tree. If there are one or more counterexamples, that is, cases in which the sentence is true, the tree method is guaranteed to find them. If the tree method does not turn up a counterexample, that is, if all paths close, we know there are no cases in which the sentence is true. But if there are no cases in which it is true, the sentence is false in all cases; in other words, it is a contradiction. We can summarize this test by saying

To test a sentence for being a contradiction, make the sentence the first line of a truth tree. If there is an open path in the tree, this path provides a counterexample to the sentence being a contradiction. If all paths close, the sentence is a contradiction.

Applying this test to our example we get



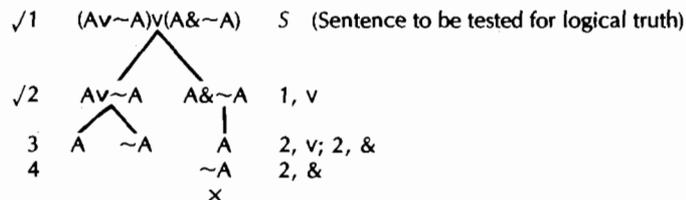
Is the sentence ' $A \equiv (\sim A \vee C)$ ' a contradiction?



Note that I have written down the result of my test, that the sentence to be tested is not a contradiction. And note how I also put down the counterexample which shows this.

A final small point about this example. In the annotation for line 4, I have listed two rules. This is because I applied the rule  $\vee$  to the disjunction on the left branch of line 3, and I separately applied the rule  $\sim \vee$  to the separate sentence of line 3 on the right branch.

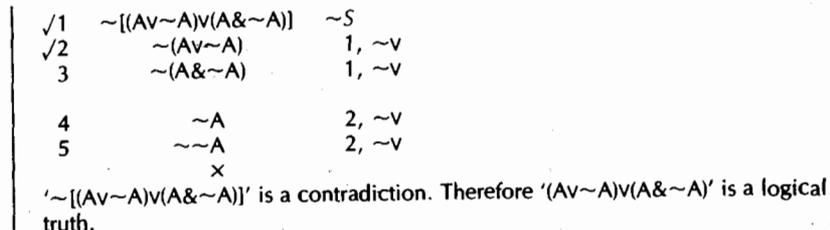
Can we use truth trees to determine whether a given sentence is a logical truth? Sometimes students propose the following test for a sentence being a logical truth: List the sentence and see if all branches remain open. Close, but no cigar! The proposed test mistakenly tells us that 'A' is a logical truth. If we make 'A' the initial line of a tree, there is nothing to do, and all branches are open. But 'A' is not a logical truth. We also get the wrong answer if we apply the proposed test to  $(A \vee \sim A) \vee (A \& \sim A)$ :



One branch of this tree does close. But the initial sentence is a logical truth, as you can tell from the fact that one of its disjuncts is a logical truth.

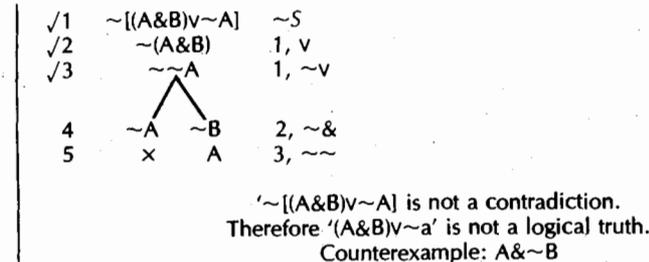
However, there is a very simple way to use the tree method to test for a logical truth. Just use the test for contradictions! How? A sentence is a logical truth just in case it is true in all cases. But a sentence is true in all cases just in case its negation is false in all cases. So a sentence is a logical truth if and only if its negation is a contradiction. Suppose, now, that I ask you whether a sentence is a logical truth, for example, the sentence of the very last example. Take the negation of the sentence. Determine

whether this negation is a contradiction. If the negation is a contradiction, then the original sentence was a logical truth. If the negation is not a contradiction, then the original sentence was not a logical truth:



Notice that this last tree is an example of a completed tree in which not all compound sentences have been checked. I never worked on line 3 because all branches closed before I got to line 3. Once all the branches close, the tree is finished. There is no way to make all the initial sentences true. If any sentences have not been worked when all branches close, continuing and working them would give us no new information.

Let us similarly test ' $(A \& B) \vee \sim A$ ' to see whether it is a logical truth:



How should we understand the counterexample here? The case  $A \& \sim B$  ('A' true and 'B' false) is a case in which ' $\sim [(A \& B) \vee \sim A]$ ', the sentence tested for being a contradiction, is true. But ' $\sim [(A \& B) \vee \sim A]$ ' is true in a case if and only if ' $(A \& B) \vee \sim A$ ', the sentence tested for being a logical truth, is false in the case. A case in which a sentence is false proves that the sentence is not a logical truth. Such a case constitutes a counterexample to the sentence being a logical truth. So the case  $A \& \sim B$  is a counterexample to ' $(A \& B) \vee \sim A$ ' being a logical truth. Clearly, this will hold generally:

For any sentence X, any case which is a counterexample to  $\sim X$  being a contradiction will also be a counterexample to X being a logical truth.

To summarize the test for logical truth:

To test a sentence for being a logical truth, make the **negation** of the sentence the first line of a truth tree. If all the paths close, the sentence is a logical truth. An open path gives a counterexample to the original sentence being a logical truth.

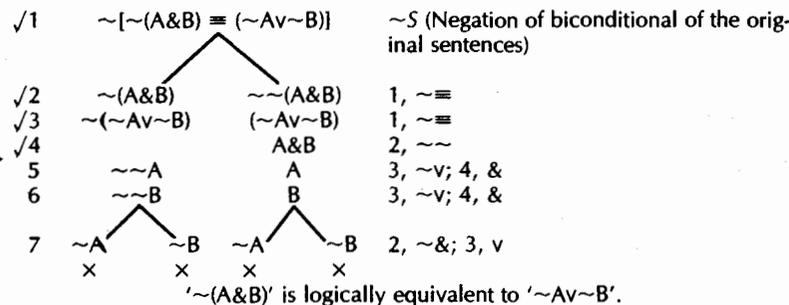
This summary makes no mention of the intermediate role of a test for contradiction. If you do not understand the test as just summarized, go back over the last two examples and make sure you can see how they fit in with the summary I have just given.

We can do yet more with the truth tree method. Recall that two sentences are logically equivalent just in case their biconditional is a logical truth. Thus we can use the test we have just devised for logical truth to determine whether two sentences are logically equivalent:

To test whether  $X$  and  $Y$  are logically equivalent, test  $X \equiv Y$  for being a logical truth. If  $X \equiv Y$  is a logical truth,  $X$  and  $Y$  are logically equivalent. If  $X \equiv Y$  is not a logical truth,  $X$  and  $Y$  are not logically equivalent. A counterexample to  $X \equiv Y$  being a logical truth is also a counterexample to the logical equivalence of  $X$  and  $Y$ . That is, it is a case in which one of the two sentences,  $X$  and  $Y$ , is true and the other is false.

Can you see why a counterexample to the logical truth of  $X \equiv Y$  is also a case in which one of the two sentences,  $X$  and  $Y$ , is true and the other is false? A counterexample to  $X \equiv Y$  being a logical truth is a case in which  $X \equiv Y$  is false. But a biconditional is false if and only if one of its components is true and the other is false, that is, if  $X$  is true and  $Y$  is false or the other way around. Finally, you can see why we would call such a case a counterexample to the logical equivalence of  $X$  and  $Y$ .  $X$  and  $Y$  are logically equivalent if and only if in all cases they have the same truth value. So if we have come up with a case in which one sentence is true and the other is false, we have a case which proves by counterexample that they are not logically equivalent.

To illustrate this method of determining logical equivalence, I will use it to verify one of De Morgan's rules. To show: ' $\sim(A \& B)$ ' is logically equivalent to ' $\sim A \vee \sim B$ '.



This section will introduce you to one more notion:

A set of one or more sentence logic sentences is consistent if and only if there is at least one assignment of truth values to sentence letters which makes all of the sentences true.

The truth tree method applies immediately to test a set of sentences for consistency. Before reading on, see if you can figure out this test for yourself.

The tree method works by finding a case in which all initial sentences on a tree are true, if there is such a case. So, to determine whether the sentences in a given set are consistent, list the sentences as the initial part of a tree. If there is a case in which all of these sentences are true together, the tree method will find it. Such a case constitutes what logicians call a *Model*, which shows the initial sentences to constitute a consistent set. If the tree closes, the set of sentences has no model and is *Inconsistent*:

A *Model* of a set of sentence logic sentences is an assignment of truth values to sentence letters which makes all of the sentences in the set true.

To test a finite set of sentences for consistency, make the sentence or sentences in the set the initial sentences of a tree. If the tree closes, there is no assignment of truth values to sentence letters which makes all the sentences true (there is no model), and the set is inconsistent. An open branch gives a model and shows the set to be consistent.

To make the statement of the test correct, I have had to use the notion of a *Finite Set* of sentences, that is, a set or collection of a limited, as opposed to an unlimited, number of sentences. You do not need to understand the distinction between a finite as opposed to an infinite set to understand anything in Volume I and Volume II, Part I of this text. But you may explore the idea and its implications for consistency by working through exercise 9-7.

It's time to practice these new applications of truth trees.

**EXERCISES**

9-3. Use the truth tree method to determine which of the following sentences are logical truths. Show your completed trees. If a sentence is not a logical truth, give the counterexample or counterexamples which show this.

- a)  $\sim(\sim F \& G) \vee \sim(F \& G)$
- b)  $[H \supset (O \supset N)] \supset [(H \& O) \supset N]$
- c)  $[\sim S \& (G \vee K)] \vee [\sim G \supset (S \vee K)]$
- d)  $[(F \vee G) \supset (P \& Q)] \supset (\sim Q \supset \sim F)$
- e)  $[L \vee (M \& N)] \supset [(L \vee M) \& N]$

- f)  $[(K \& M) \& P] \supset [(\sim K \& P) \vee (K \vee \sim M)]$   
 g)  $[L \supset (M \supset N)] \supset [(L \supset M) \supset (L \supset N)]$   
 h)  $[(S \vee T) \supset F] \supset \{[(F \vee G) \supset H] \supset (S \supset H)\}$   
 i)  $(I \& \sim J) \vee [(J \& K) \vee \sim (K \& I)]$   
 j)  $\{[C \& (A \vee D)] \vee \sim (C \& F)\} \vee \sim (A \& \sim G)$

9-4. Use the truth tree method to determine which of the following sentences are contradictions. Show your completed trees, and for each sentence you determine not to be a contradiction, give the counterexample or counterexamples which show this.

- a)  $(A \& B) \& (\sim A \vee \sim B)$   
 b)  $(F \vee G) \& (\sim F \vee \sim G)$   
 c)  $(F \vee G) \equiv (\sim F \& \sim G)$   
 d)  $[I \vee (J \& K)] \supset [(I \vee J) \& K]$   
 e)  $[(H \& F) \supset C] \& \sim [H \supset (F \supset C)]$   
 f)  $\sim \{[B \& (M \vee \sim P)] \supset [P \supset (\sim M \vee \sim B)]\}$   
 g)  $(A \equiv B) \equiv (\sim A \equiv B)$   
 h)  $[(\sim F \vee Q) \vee (P \& Q)] \& \sim (\sim Q \supset \sim F)$   
 i)  $[K \supset (D \supset P)] \& [(\sim K \vee D) \& \sim (K \supset P)]$   
 j)  $[(\sim G \vee Q) \& (K \supset G)] \& \sim (P \vee \sim K)$

9-5. Use the information suggested in exercise 4-3 to state a new truth tree test for logical equivalence. Comment on the relation between this test and the test given in the text.

9-6. Use the truth tree method to determine which of the following pairs of sentences are logically equivalent. You may use either the test given in the text or the closely related test you discovered in exercise 9-4. Show your completed trees, and when you find a pair which is not logically equivalent, give the counterexample or counterexamples which show this.

- a)  $A \supset \sim A$  and  $\sim A$   
 b)  $\sim (I \vee J)$  and  $\sim I \& \sim J$   
 c)  $M \vee \sim H$  and  $\sim M \supset H$   
 d)  $\sim (F \& P)$  and  $\sim F \& \sim P$   
 e)  $(D \& N) \supset J$  and  $D \supset (N \supset J)$   
 f)  $(I \supset Q) \supset D$  and  $I \supset (Q \supset D)$   
 g)  $L \& (S \vee T)$  and  $(L \& S) \vee (L \& T)$   
 h)  $H \vee \sim (\sim P \vee \sim Q)$  and  $(H \vee P) \& (H \vee Q)$

9-7. Consider the following definition:

- (C1) A sentence of sentence logic is consistent if and only if it is not a contradiction.

a) Are all logical truths consistent according to definition (C1)? Explain why or why not.

b) Show that a sentence logic sentence is consistent according to definition (C1) if and only if there is at least one assignment of truth values to sentence letters which makes the sentence true.

If you go on in your study of logic, the more general notion of consistency already given in the text will turn out to be very important:

- (C2) A set of one or more sentence logic sentences is consistent if and only if there is at least one assignment of truth values to sentence letters which makes all of the sentences true.

c) Show that a set of sentences is consistent according to definition (C2) if and only if the conjunction of all the sentences in the set is consistent according to definition (C1).

This last problem seems to show that the two definitions of consistency come to the same thing. Why, then, did I say that the second definition is more general? Actually, there is something not quite right about exercise (c). To see what this is, you need to understand the difference between a finite and an infinite set of sentences. A finite set of sentences has some definite number of sentences in it, such as 2, or 47, or 1,007,859. In an infinite set of sentences the list of sentences goes on without end. There is no integer which gives you the number of sentences in an infinite set.

d) Here is an example of an infinite set of sentences:

$\sim A, \sim \sim A, \sim \sim \sim A, \sim \sim \sim \sim A, \sim \sim \sim \sim \sim A, \dots$

The first sentence is ' $\sim A$ '. The second sentence is ' $\sim \sim A$ '. The third, fourth, fifth, and further sentences are ' $A$ ' preceded by 3, 4, 5, and further negation signs, so that the list goes on forever. Question: Is this set consistent?

The difficulty with exercise (c) is that it makes sense only if you assume that the set of sentences is finite. For if the set is infinite, there is no sentence which is the conjunction of all its members. This is because in sentence logic, all sentences are finite in length.

Now you can see why definition (C2) is more general than (C1). (C2) gives sense to the consistency of an infinite set of sentences. The two definitions really come to the same thing, but only for **finite** sets of sentences.

e) Describe a consistent infinite set of sentences.

f) Use the truth tree method to test the following sets of sentences for consistency. In each case, show your tree. Write next to your tree whether the set is consistent or inconsistent, and when consistent, give all the models for the set which the truth tree provides.

- f1)  $P \vee S, P \supset S$
- f2)  $(\sim F \supset S) \supset F, \sim F, S$
- f3)  $K \& [(\sim K \& H) \vee H]$
- f4)  $N \vee B, N \vee \sim B, \sim N \vee B$

From a logical point of view, you should really think of the truth tree method as a method for testing a set of sentences for consistency. This general method then has more specific applications, such as testing an argument for validity. This is because

An argument is valid if and only if the set comprised by the argument's premises and the negation of its conclusion is inconsistent.

We say the same thing in other words by saying that an argument is valid if and only if the negation of the conclusion is inconsistent with (the set of) the argument's premises.

g) Explain why the last offset statement is correct.

Some textbooks first present the truth tree method as a test of consistency and then apply it to argument validity. I introduced trees as a test for argument validity because I wanted to motivate the introduction of trees with something you already know about, namely, arguments. It is initially hard for many students to understand the interest in consistency and inconsistency, but these notions will become very important in Volume II, Part II of the text.

#### CHAPTER SUMMARY EXERCISES

Here are the important terms and ideas from this chapter. Write your explanations for them as usual. This list repeats some terms from previous chapters.

- a) Main Connective
- b) Logical Truth
- c) Truth Tree Test for Logical Truths
- d) Contradiction
- e) Truth Tree Test for Contradictions
- f) Logical Equivalence
- g) Truth Tree Test for Logical Equivalence
- h) Consistency
- i) Model
- j) Infinite Set of Sentences
- k) Truth Tree Test for Consistency of a Finite Set of Sentences