

# Soundness and Completeness for Sentence Logic Trees

## 12

### 12-1. PRELIMINARIES

This chapter will explain the soundness and completeness of sentence logic for the tree method. Section 12-2 gives an informal statement which you will be able to follow without having studied more than the first short section of chapter 11, on mathematical induction. Section 12-3 gives full details.

Before getting started, I want to make a general point which will be useful in discussing both trees and derivations. I am going to make a statement which uses a new bit of notation. 'U' indicates set union. That is,  $Z \cup W$  is the set consisting of all the members of the set  $Z$  together with the members of the set  $W$ . Also, if  $X$  is a sentence,  $\{X\}$  is the set which has  $X$  as its only member. Now, to say that the  $Z \supset X$  is valid, that is, that  $Z \vDash X$ , is to say that every interpretation which makes all the sentences in  $Z$  true also makes  $X$  true. Keep in mind that  $X$  is true in an interpretation iff  $\sim X$  is false in that interpretation. Consequently

L1:  $Z \vDash X$  iff  $Z \cup \{\sim X\}$  is inconsistent.

(The 'L' in 'L1' stands for 'lemma'. A lemma is a statement which may not be of great interest in itself but which we prove because it will be useful in proving our main results.)

L1 shows that validity of an argument comes to the same thing as the

inconsistency of a certain set of sentences, namely, the premises and negation of the conclusion of the argument. You will soon see that L1's equivalent formulation of validity provides a particularly convenient way to study soundness and completeness.

#### EXERCISE

12-1. Prove L1.

### 12-2. SOUNDNESS AND COMPLETENESS OF THE TREE METHOD: INFORMAL STATEMENT

Soundness and completeness tell us that there is an exact correspondence between a semantic concept—validity—and a corresponding syntactic concept—proofs. Let's be explicit about what counts as a proof in the tree method: Given some premises and a conclusion, a tree method proof is a closed tree (a tree with all its branches closed) which has the premises and negation of the conclusion as its initial sentences. Closed trees are the syntactic objects which need to correspond to the semantic concept of validity. So proving soundness and completeness for the tree method means proving that we have the right sort of correspondence between validity and closed trees.

To become clear on what correspondence we need, let's go back to the way I introduced the tree method. I said that, given an argument,  $Z \supset X$ , the argument is valid just in case it has no counterexamples, that is, no interpretations in which the premises,  $Z$ , are all true and the conclusion,  $X$ , is false. I then went on to develop truth trees as a method of looking for counterexamples, a way which is guaranteed to find a counterexample if there is one. If the whole tree closes, there are no counterexamples, and we know the argument is valid. But a closed tree is what counts as a proof. So if there is a proof, the argument is valid. If you look back at definition D3 in chapter 10, you will see that this is what we call soundness.

On the other hand, if there is an open branch (and so no proof), there is a counterexample, and thus the argument is invalid. A little thinking indicates that the last statement is just completeness: "If no proof, then invalid" comes to the same as "If valid, then there is a proof," which is completeness, as defined by D4 in chapter 10. I have used the law of contraposition:  $X \supset Y$  is logically equivalent to  $\sim Y \supset \sim X$ .

The first time through, this argument is bound to seem very slick. It is

also incomplete: I have yet to prove that the truth tree method is guaranteed to find a counterexample if there is one.

To sort all of this out more carefully, we need to examine the connection between a counterexample and lemma L1. A counterexample to the argument  $Z \mathbf{X}$  is just an interpretation which shows the set of sentences,  $Z \cup \{\sim \mathbf{X}\}$  to be consistent. (Remember that such an interpretation is called a model of the set of sentences.) Now, look at lemma L1, and you will start to see that all this talk about counterexamples is just another way of saying what lemma L1 says.

**EXERCISE**

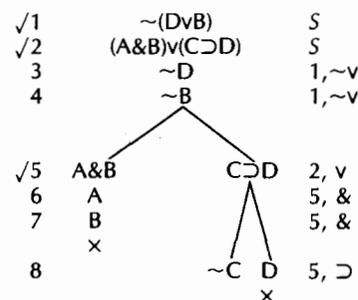
12-2. Show that lemma L1 is equivalent to the statement that an argument is valid iff it has no counterexamples.

Lemma L1 tells us that we can forget about validity and talk about consistency and inconsistency instead. Indeed, conceptually, the tree method is really a method for determining whether the initial sentences on a tree form a consistent set. It is a method which is guaranteed to find a model for a tree's initial sentences if there is one, thereby showing the set of sentences to be consistent. Conversely, if a set is inconsistent, it has no model, and a tree starting with the sentences in the set is bound to close.

The real work we have to do is to show that the tree method is guaranteed to find a model for a set of sentences if the set has a model. We'll worry later about connecting this up with validity—lemma L1 assures us that we will be able to do so. For now, we will connect the semantic concept of a model with the syntactic concept of an open branch. Remember that an open branch always represents an interpretation in which all sentences on the branch are true. Hence, if there is an open branch, there is an interpretation in which all the sentences on the branch, including the tree's initial sentences, are true.

Here is how we proceed: We will show that a finite set of sentences is consistent if and only if we always get an open branch on a finished tree which starts from the sentences in the set. Equivalently, a set is inconsistent if and only if we always get a closed tree if we start from the sentences in the set. This gives us the connection between a syntactic concept—open and closed trees—and a semantic concept—consistency and inconsistency. Lemma L1 tells us we will be able to connect the latter with validity and invalidity.

To keep track of how we will carry out this program, let's talk about it in terms of an example, say, the tree which results from using as initial sentences the sentences in the set  $\{\sim(D \vee B), (A \& B) \vee (C \supset D)\}$ :



We must first show that tree method is what I will call *Downwardly Adequate*. This means that the tree method applied to a consistent set of sentences always results in at least one open branch. Why is the tree method downwardly adequate? Remember that the rules are written so that when a rule is applied to a sentence, the rule instructs you to write, on separate branches, all the minimally sufficient ways of making the original sentence true. In effect, this means that, for any assignment of truth values to sentence letters which makes the original sentence true, all the sentences of at least one of the resulting stacks of sentences will be true also for the same assignment of truth values.

This fact is easiest to grasp by example. In applying the rule  $\vee$  to line 2, we produce two branches, on line 5. Suppose that we have an assignment that makes line 2 true. This can only be because it makes at least one of the two disjuncts true. But then, on this assignment, at least one of the sentences on the two branches of line 5 will be true for the same assignment.

Or consider application of the rule  $\sim \vee$  to line 1, and suppose that we have a truth value assignment which makes line 1 true. A negated disjunction can be true only if both disjuncts are false. Consequently, an assignment which makes line 1 true will make lines 3 and 4 true.

As I introduced the rules, I made sure that they are all, as in these two examples, what I will call *Downwardly Correct*. In outline, the downward correctness of the rules works to show that the tree method is downwardly adequate as follows: Suppose that the initial sentences on a tree are consistent, so that there is an interpretation,  $I$ , which assigns truth values to sentence letters making all the initial sentences true. Now apply a rule to one of the sentences. The downward correctness of the rules means that applying the rule will result in at least one branch on which all the new sentences are true in  $I$ . Of course, all the former sentences along this branch were true in  $I$ . So at this stage we have at least one branch along which all sentences are true in  $I$ . Now we just do the same thing over again: Apply one of the rules to a sentence on this branch. We get at least one extension of the branch on which all sentences are true in  $I$ .

This process will eventually come to an end because each application of a rule produces shorter sentences. At the end we have at least one branch on which all the sentences are true in  $I$ . But this branch must be open. Since all the sentences along the branch are true in  $I$ , no sentence and its negation can both appear on the branch! In sum, if the original sentences are consistent, there will be an open branch.

We are half done. We must still show that the tree method is *Upwardly Adequate*, that is, that if there is an open branch, then the set of initial sentences is consistent. So now let us suppose that we have a tree with an open branch. Since an open branch never has both a sentence and its negation,  $I$  can consistently assign the truth value  $t$  to all atomic sentences on the branch and the truth value  $f$  to all those atomic sentences whose negations occur on the branch. Call this assignment  $I$ .  $I$  will also make the longer sentences on the branch true.

Look, for instance, at the open branch in the last example. Reading up from the bottom, this branch specifies the assignment 'C', 'B', and 'D' all false. Call this assignment  $I$ . If 'C' is false, that is, if  $\sim C$  is true in  $I$ , then 'C $\supset$ D' is true in  $I$ . In turn, 'C $\supset$ D' being true in  $I$  will make line 2, '(A&B) $\vee$ (C $\supset$ D)' true in  $I$ . Likewise, lines 3 and 4, ' $\sim D$ ' and ' $\sim B$ ', true in  $I$  will make line 1, ' $\sim(D\vee B)$ ', true in  $I$ .

All the rules have the property just used, called *Upward Correctness*: If  $I$  makes true the sentence or sentences which a rule produced from a previous sentence,  $I$  makes that previous sentence true also. Upward correctness will apply to any open branch in any tree just as it did in the example. Choose an interpretation,  $I$ , as the one which makes all the atomic sentences on the open branch true and all the negated atomic sentences false. Apply upward correctness again and again. You can see that, finally, all the sentences along the open branch are true in  $I$ . So the open branch provides an interpretation,  $I$ , which makes all the sentences along the branch true, including the initial sentences. So if there is an open branch there is a model for the initial sentences, which is to say that the initial sentences form a consistent set, which is just what we mean by upward adequacy.

Let's pull the threads together. The tree method is downwardly adequate. That is, if the initial sentences are consistent, then there is an open branch. By contraposition, if there is no open branch, that is, if there is a proof, then the initial sentences form an inconsistent set. Lemma 1 tells us that then the corresponding argument is valid. This is soundness.

The tree method is also upwardly adequate. If there is an open branch, and so no proof, then the initial set of sentences is consistent. By contraposition, if the set of initial set of sentences is inconsistent, then there is a proof. Lemma 1 then connects the inconsistency with validity: If the corresponding argument is valid, there is a proof. This is completeness.

If you are starting to see how soundness and completeness work for trees, this section is doing its job. Doing the job fully requires further precision and details, presented in the next section. If in the next section you start to feel lost in a forest of definitions (as I often do) reread this section, which contains all the concepts. Reviewing the intuitively presented concepts will help you to see how the following formal details fit together.

### 12-3. SOUNDNESS AND COMPLETENESS FOR SENTENCE LOGIC TREES: FORMAL DETAILS

In this section I am going to make a very natural simplifying assumption: I will restrict discussion to **finite** sets of sentences  $Z$ . This restriction is natural because intuitively we think of arguments as only having finitely many premises anyway. Generalization to the case of infinite sets of sentences involves a complication which would only distract us from the main line of argument. Chapter 14 will take care of the generalization.

For precision and efficiency of statement, we need the following definitions:

D8: A *Minimal Sentence* is a sentence which is either atomic or the negation of an atomic sentence.

D9: A truth tree is *Finished* iff each branch is either closed or has all applicable rules applied to all nonminimal sentences.

D10: 'Tr', 'Op', and 'Cl' are predicates of the metalanguage (abbreviations in English) which are defined as

- a)  $\text{Tr}(T, Z)$  iff  $T$  is a finished tree with all the sentences in  $Z$  its initial sentences.
- b)  $\text{Op}(T)$  iff the tree  $T$  is open, that is, if  $T$  has an open branch.
- c)  $\text{Cl}(T)$  iff  $\sim\text{Op}(T)$ , that is, if  $T$  is closed, that is, if all of  $T$ 's branches are closed.

A proof of  $ZX$  is just a closed tree which starts with sentences in  $Z$  and the sentence  $\sim X$ . Expressed with our abbreviations, this is

D11: A tree,  $T$ , is a *proof of X from Z* iff  $\text{Tr}(T, Z \cup \{\sim X\})$  and  $\text{Cl}(T)$ .

Next, recall that  $Z \vdash X$  just means that **there exists** a proof of  $X$  using premises in  $Z$ , where here a proof just means a tree as described in D11. So applying D11 to the definition of  $\vdash$ , we have

L2: For finite sets,  $Z$ ,  $Z \vdash X$  iff  $(\exists T)[\text{Tr}(T, Z \cup \{\sim X\}) \ \& \ \text{Cl}(T)]$ .

Of course, throughout this section ‘ $\vdash$ ’ means  $\vdash_{\tau}$ , that is, derivability according to the tree method.

**EXERCISE**

12–3. In chapter 10 I specified that  $Z \vdash X$  means that there is a proof of  $X$  using any number of sentences in  $Z$ , but not necessarily all of them. (I did this to accommodate the eventual generalization to infinite sets.) But D11 defines  $T$  as being a proof of  $X$  from  $Z$  in terms of  $\text{Tr}(T, Z \cup \{\sim X\})$ , which specifies a tree,  $T$ , which has **all** the sentences of  $Z \cup \{\sim X\}$  as its initial sentences.

Prove L2, taking care to deal with this apparent difficulty. Use the fact that L2 is stated with the existential quantifier, ‘ $(\exists T)$ ’.

Now remember how we used L1 to show that we could exchange the study of validity and invalidity for the study of the consistency and inconsistency of a certain set of sentences, namely, the premises together with negation of the conclusion. Our next step is to connect the consistency of this set with the syntactic notion of an open branch. We do this with the idea of upward and downward adequacy of the tree method. Downward adequacy says that if the set  $Z$  is consistent, that is, if there is a model for  $Z$ , then the tree starting from  $Z$  has an open branch. Using definitions D5 and D6, this becomes

D12: The tree method is *Downwardly Adequate* iff for all finite, nonempty sets of sentences  $Z$ , if  $(\exists I)\text{Mod}(I, Z)$ , then  $(\forall T)[\text{Tr}(T, Z) \supset \text{Op}(T)]$ .

Upward adequacy is the converse: If there is an open branch, the initial set is consistent:

D13: The tree method is *Upwardly Adequate* iff for all finite, nonempty sets of sentences  $Z$ , if  $(\forall T)[\text{Tr}(T, Z) \supset \text{Op}(T)]$ , then  $(\exists I)\text{Mod}(I, Z)$ .

A detail in D12 and D13 requires comment. If we start a tree with the sentences in  $Z$ , we can come up with more than one tree because we can apply the rules in different orders. So when I give a formal definition of upward and downward adequacy, I must make a choice whether to define these in terms of **all** open trees starting from  $Z$  or **some** open tree starting from  $Z$ .

In terms of the proof of upward and downward adequacy, I could do either because, in essence, the proof will show that, for a given set of initial sentences, one tree is open iff all are. I choose to define upward and downward adequacy in terms of all open trees for the following rea-

son: When we connect adequacy with soundness and completeness, I will be taking a converse. This will introduce a negation sign, and when the negation sign gets pushed through the quantifier, ‘all’ turns into ‘some’. At that point I will be talking about “some closed tree”. That is just what we will need to get a smooth fit with derivability, which is defined in terms of “there is **some** proof”, where a proof is just a closed tree. If I had defined upward and downward adequacy in terms of some instead of all open trees, it would be a mess to make the connection with soundness and completeness.

**EXERCISE**

12–4. Assume (as we will prove shortly) that if a tree has at least one open branch, then the initial sentences of the tree form a consistent set. Also assume downward adequacy. Prove that for all the finished trees starting from the same set of initial sentences, one is open iff all are.

The next step is to show that upward adequacy is equivalent to soundness and downward adequacy is equivalent to completeness. The connection will not sink in unless you do the work! But I will break the job down into several steps.

First we define a version of soundness and completeness for the tree method:

D3’: The tree method is *Sound* iff for all finite, nonempty sets of sentences  $Z$ , if  $(\exists T)[\text{Tr}(T, Z) \ \& \ \text{Cl}(T)]$ , then  $(\forall I)\sim\text{Mod}(I, Z)$ .

D4’: The tree method is *Complete* iff for all finite, nonempty sets of sentences  $Z$ , if  $(\forall I)\sim\text{Mod}(I, Z)$ , then  $(\exists T)[\text{Tr}(T, Z) \ \& \ \text{Cl}(T)]$ .

Now it is not hard to prove that downward adequacy is soundness and upward adequacy is completeness in the form of four new lemmas:

L3: The tree method is sound according to D3 iff it is sound according to D3’.

L4: The tree method is complete according to D4 iff it is complete according to D4’.

L5: The tree method is sound according to D3’ iff it is downwardly adequate.

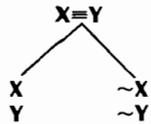
L6: The tree method is complete according to D4’ iff it is upwardly adequate.

**EXERCISES**

- 12-5. Prove lemmas L3 and L4. You will need to use lemmas L1 and L2.  
 12-6. Prove lemmas L5 and L6. You will need to use contraposition and the laws of logical equivalence for negated quantifiers as laws applied to statements in the metalanguage.

We have reduced the problem of proving soundness and completeness to that of proving that the tree method is downwardly and upwardly adequate, which the last section indicated we would do by appealing to the downward and upward correctness of the rules. Some further terminology will enable us to state rule correctness more clearly.

When we apply a truth tree rule to a sentence, the rule instructs us to write one or two branches and on each branch one or two new sentences. For example, the  $\equiv$  rule is



We will call the sentence to which the rule is applied,  $X \equiv Y$  in the example, the *Input Sentence*. The rule instructs you to write one or two lists of sentences (each "list" containing one or two sentences). We will call each such list an *Output List*. In the example,  $X, Y$  is one output list and  $\sim X, \sim Y$  is the second output list. The rule



has only one output list, namely,  $X, Y$ . In summary

**D14:** The sentence to which a tree method rule is applied is called the *Input Sentence*. The sentence or (sentences) along one branch which the rule directs you to write is (are) called an *Output List of Sentences*.

Here is what we must require of a correct truth tree rule. Suppose that  $I$  give you an interpretation (an assignment of truth values to sentence

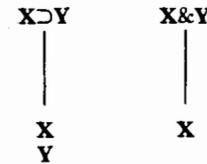
letters) which makes true the input sentence of a rule. Then that same interpretation must make true all the sentences on at least one (but perhaps not all) output lists. This is downward correctness. And suppose  $I$  give you an interpretation which makes all the sentences on one output list true. Then that same interpretation must make the input sentence true. This is upward correctness.

**D15:** A tree method rule is *Downwardly Correct* iff any interpretation which makes the input sentence true also makes true all the sentences of at least one output list.

**D16:** A tree method rule is *Upwardly Correct* iff any interpretation which makes all the sentences of one output list true also makes the input sentence true.

**EXERCISES**

- 12-7. Show that all of the truth tree rules for sentence logic are downwardly and upwardly correct.  
 12-8. Consider the following two proposed truth tree rules:



Determine which of these is downwardly correct and which is upwardly correct. In each case show correctness or failure of correctness.

We are now ready to prove

**T1:** The truth tree method for sentence logic is downwardly adequate.

(The 'T' stands for 'theorem'.) Suppose we are given a finite nonempty set of sentences,  $Z$ , and a tree,  $T$ , which has the sentences of  $Z$  as its initial sentences. Now suppose that there is a model,  $I$ , of the sentences in  $Z$ . What we will do is to look at successively larger initial segments of one branch of  $T$  and show that all these initial segments of the branch are open.

Start with just the sentences in  $Z$ , that is, the initial sentences of  $T$ . This initial segment of a branch must so far be open. Why? Well, a branch

closes only if it contains both a sentence and the negation of that same sentence. But  $Z$  can't contain a sentence and its negation. This is because there is a model,  $I$ , of all the sentences in  $Z$ . That is,  $I$  makes **all** the sentences in  $Z$  true. But no sentence and its negation can both be true in the same interpretation! If  $I$  makes one sentence true, it makes its negation false. So far we have an initial segment—let's call it the first segment—of a branch, all the sentences of which are true in  $I$ , and which consequently is (so far) open.

Next, in constructing the tree  $T$ , we apply a rule to one of the sentences in this first initial segment of our so far open branch. The input sentence for this rule is true in  $I$ . By the downward correctness of the rules, there will be at least one output list all the sentences of which are true in  $I$ . Pick one such output list (say, the left one if there are more than one). Look at the extension of the first segment of our branch extended by this output list. Call this extension the second initial segment. This second segment now has all its sentences true in  $I$ .

You can see the handwriting on the wall. We just do the same thing over and over again. At the  $n$ th stage we start with a branch all the sentences of which are true in  $I$ . The tree grows by application of some rule to some sentence on the  $n$ th initial segment. Downward correctness guarantees that at least one output list will have all its sentences true in  $I$  also. We pick the leftmost such output list as the extension of the  $n$ th initial segment to the  $n + 1$ st initial segment. Then the  $n + 1$ st initial segment has all its sentences true in  $I$ , and we can start all over again.

In a sentence logic tree, the sentences get shorter with each application of a rule, so this process must eventually end. When it does, we have a branch all the sentences of which are true in  $I$ . For exactly the same reason that the first initial segment must be open, this final branch must be open also: All its sentences are true in  $I$ , and no sentences and its negation can both be true in the same interpretation.

#### EXERCISE

12–9. Formulate the foregoing argument sketch into an explicit use of mathematical induction to prove T1. There are many correct ways to apply induction. For example, begin by supposing that you are given a finite, nonempty set of sentences,  $Z$ , a model  $I$  of  $Z$ , and a finished tree,  $T$ , with initial sentences  $Z$ . Break the tree up into stages: The  $n$ th stage of the tree includes all lines written down in the first through  $n$ th application of a rule. Your inductive property will be: There is a branch through the  $n$ th stage of the tree all the sentences of which are true in  $I$ . Or you can similarly organize the inductive property around the number of lines to be checked: The

first line to be checked, the first and second lines to be checked, and so on. Be sure to show explicitly how the results from the induction establish downward adequacy.

I have suggested a formulation for this proof which I hope you will find to be relatively intuitive, but the logical form of the suggested proof is actually a bit convoluted. In this formulation you use both universal introduction and induction. That is, for an arbitrary, finite, nonempty set  $Z$ , model  $I$  of  $Z$ , and tree  $T$  with initial sentences in  $Z$ , you show how induction gives the desired result in that case. Then, since you have assumed nothing else about the  $Z$ ,  $I$ , and  $T$ , what you have shown is true for all such  $Z$ ,  $I$ , and  $T$ . In addition, the induction is a finite induction. In a specific case it runs only from the first through the number of stages in the tree in question.

Logicians prefer a more abstract but “pure” way of doing this kind of problem. In the inductive step you assume that in any tree with  $n$  stages (or  $n$  checked sentences) and interpretation  $I$  which makes all initial sentences true, there is a path all the sentences of which are true in  $I$ . You then use downward rule correctness to show that the same is true in any  $n + 1$ -stage tree. To do this you consider an arbitrary  $n + 1$ -stage tree and the  $n$ -stage tree (or trees) which result by deleting the first sentence to which a rule was applied in the original  $n + 1$ -stage tree. The downward rule correctness of the applied rule shows that if the inductive hypothesis holds of the subtree, it holds of the full  $n + 1$ -stage tree.

But I will leave the details to you and your instructor!

Let's turn to

T2: The truth tree method for sentence logic is upwardly adequate.

The proof works similarly to that for downwardly adequate, differing in that we use upward correctness of the rules and we look at successively longer and longer sentences on a branch instead of successively longer and longer initial segments of a branch.

Suppose we are given a tree with an open branch. Take one open branch (say, the leftmost). Because this branch is open, and so has no sentence and its negation, we can consistently assign the truth value  $t$  to all the atomic sentence letters which appear on the branch and the truth value  $f$  to all atomic sentence letters the negation of which appear on the branch. This constitutes an interpretation  $I$ —an assignment of truth values to sentence letters. We are going to show that all the sentences along this branch are true in  $I$ .

By the *Length* of a sentence let us understand the total number of con-

nectives and sentence letters that appear in the sentence. So far, all minimal sentences along the branch are true in  $I$ —that is, all sentences of length 1 or 2. Now, consider any sentence along the branch (if there are any) of length 3. When a rule was applied to such a sentence, the rule produced an output list the sentences of which are each shorter than the input sentence; that is, each has fewer total connectives plus sentence letters. (You should check this.) But all such shorter sentences of the branch, that is, sentences of length 1 or 2, are already known to be true in  $I$ . Upward rule correctness then tells us that the sentence of length 3 is true in  $I$ . The same goes for any length 3 sentence on the branch. So now we know that all sentences of length 1, 2, and 3 on the branch are true in  $I$ .

Again, you can see how this will go: We do the same thing over and over again. At stage  $n$  we know that all sentences of the branch of length  $n$  or less are true in  $I$ . Consider any sentence of length  $n + 1$ . The rule applied to it produced shorter sentences, already known to be true in  $I$ . By upward correctness of the applied rule, the sentence of length  $n + 1$  is then also true in  $I$ . The same goes for any sentence of length  $n + 1$  on the branch, so that we now have shown that all of the branch's sentences of length  $n + 1$  are true in  $I$ . Ultimately, the process shows that all the sentences in the branch are true in  $I$ . This includes the initial sentences, which are the initial sentences of the tree.

#### EXERCISE

12–10. Formulate the foregoing argument sketch into an explicit inductive argument. That is, given a tree and an open branch on the tree, show that there is an interpretation which can be shown by induction to make all sentences (and hence the initial sentences) along the branch true.

Comments exactly parallel to those on your proof of T1, about the logical “purity” of the proof, also apply here. Just as for T1, one can also do the induction on the “size” of the tree. In the inductive step, you assume that all open trees with no more than  $n$  checked sentences have the desired characteristic—that open paths represent interpretations which make all the sentences on the path true—and you then use upward rule correctness to show that all trees with  $n + 1$  checked sentences also have this characteristic. In outline, the idea is that any tree with  $n + 1$  checked sentences has one or more subtrees with no more than  $n$  checked sentences—namely, the tree or trees obtained by deleting the first checked sentence in the original tree. You then apply the inductive hypothesis assumed to hold for the shorter trees.

We have shown that, given some tree with an open branch, there is an interpretation,  $I$ , in which all of the tree's initial sentences are true. How does this show upward adequacy? Suppose we are given a finite, non-empty set of sentences,  $Z$ . Assume the antecedent in the statement of upward adequacy. That is, assume that any tree starting from  $Z$  is open. There is always at least one tree,  $T$ , starting from  $Z$ . Since all such trees are being assumed to be open,  $T$  is open, that is,  $T$  has an open branch. But in the previous paragraphs we have shown that this open branch provides an interpretation in which all initial sentences of  $T$ , that is, all the sentences in  $Z$ , are true.

We have now completed the proof of T2.

T1 and T2, with the help of lemmas L3, L4, L5, and L6, complete our proof of soundness and completeness for the tree method. As you can check in a moment, T1, L3, and L5 immediately give

T3: The tree method for sentence logic is sound.

T2, L4, and L6 immediately give

T4: The tree method for sentence logic is complete.

#### EXERCISES

12–11. This exercise makes precise the work you did informally in exercises 10–14 and 10–15. Recall that when I refer to consistency and inconsistency without qualification, I always mean semantic consistency and inconsistency. We want a notion of *Syntactic Consistency and Inconsistency*, that is, a syntactic notion which will work as a test for semantic consistency and inconsistency. These are

D17:  $Z$  is *Syntactically Consistent* iff  $(\forall T)[\text{Tr}(T, Z) \supset \text{Op}(T)]$ .

D18:  $Z$  is *Syntactically Inconsistent* iff  $(\exists T)[\text{Tr}(T, Z) \& \text{Cl}(T)]$ .

(Throughout this problem, be sure to assume that  $Z$  is finite and nonempty.)

- Show that a set of sentences is syntactically consistent according to D17 iff it is not syntactically inconsistent according to D18.
- Show that  $Z$  is syntactically consistent iff  $Z \nVdash A \& \sim A$ .
- Show that  $Z$  is syntactically inconsistent iff  $Z \vdash A \& \sim A$ .
- Show that  $Z$  is syntactically inconsistent iff for any  $X$ ,  $Z \vdash X$ .
- Reexpress lemma L2 and definitions D12, D13, D3', and D4' in terms of semantic and syntactic consistency.

**CHAPTER CONCEPTS**

To check your understanding of this chapter, make sure that you understand all of the following:

- a) Input Sentence of a Rule
- b) Output Sentence of a Rule
- c) Downward Rule Correctness
- d) Upward Rule Correctness
- e) Downward Adequacy
- f) Upward Adequacy
- g) Minimal Sentence
- h) Finished Tree
- i)  $\text{Tr}(T, Z)$
- j)  $\text{Op}(T)$
- k)  $\text{Cl}(T)$
- l) Tree  $T$  is a proof of  $X$  from  $Z$
- m) Syntactic Consistency
- n) Semantic Consistency