

## Open Branch Interpretations

PHIL 422

Woodbridge

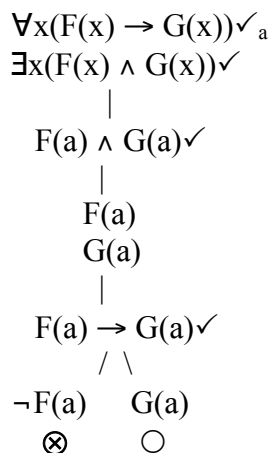
Now that you are experts at using trees, we want to start making connections between the syntactic processing our deductive apparatus implements (with the resulting syntactic assessments that yields) and semantic considerations.

As review, remember the following.

1. To test whether ' $\phi$ ' is a theorem, we test  $\vdash \phi$ , i.e., we run a tree for  $\Sigma = \{\neg\phi\}$ . If the tree for ' $\neg\phi$ ' is closed, then ' $\phi$ ' is a theorem (since the tree for ' $\phi$ ' will be open).
2. To test whether ' $\phi$ ' is an anti-theorem, we run a tree for  $\Sigma = \{\phi\}$ . If the tree for ' $\phi$ ' is closed, then ' $\phi$ ' is an anti-theorem (meaning that ' $\neg\phi$ ' is a theorem, and its tree is open).
3. To test whether ' $\phi$ ' is a neutral, we test both  $\vdash \phi$  and  $\vdash \neg\phi$ . If both trees are open, then ' $\phi$ ' is a neutral.
4. To test whether  $\Gamma \vdash \phi$ , we run a tree for  $\Sigma = \Gamma \cup \{\neg\phi\}$ . A closed tree for  $\Sigma$  means establishment (because ' $\neg\phi$ ' is incompatible with  $\Gamma$ , so if we have  $\Gamma$ , we must, syntactically speaking, have ' $\phi$ ' with it); an open tree means non-establishment.
5. To test whether a set of sentences,  $\Sigma$ , is compatible, we run a tree for the members of  $\Sigma$ . If the tree is open, the sentences in  $\Sigma$  are compatible; if it is closed, they are incompatible.
6. To test whether a pair of sentences, ' $\phi$ ' and ' $\psi$ ', are coupled, we test  $\vdash (\phi \leftrightarrow \psi)$ , i.e., we run a tree for  $\Sigma = \{\neg(\phi \leftrightarrow \psi)\}$ . They're coupled if the tree is closed, uncoupled if it's open.

In each test, the resulting tree is either open (i.e., it has **at least one** open branch) or closed (i.e., **all** of its branches are closed). If it is open, then there is at least one branch on the tree for which no sentence ' $\phi$ ' and its negation, ' $\neg\phi$ ' both occur. Each open branch will be considered **separately**. (Remember, a *literal* is an atomic sentence or a negated atomic sentence.)

Consider the tree for  $\Sigma = \{ \forall x(F(x) \rightarrow G(x)), \exists x(F(x) \wedge G(x)) \}$



The tree is open, so the sentences in this  $\Sigma$  are compatible. The open branch on this tree *contains* all of the following sentences: “ $\forall x(F(x) \rightarrow G(x))$ ”, “ $\exists x(F(x) \wedge G(x))$ ”, “ $(F(a) \wedge G(a))$ ”, “ $F(a)$ ”, “ $G(a)$ ”, and a second example of “ $G(a)$ ”. An ***open branch interpretation*** (an *OBI*) makes all of the sentences on the open branch true. Specifically, it makes all of the sentences in  $\Sigma$  true, which would show that  $\Sigma$  has the *semantic* relation of being consistent, as well as the *syntactic* relation of being compatible.

We construct an OBI by following a step-by-step procedure involving these steps:

- Step 1:** Identify ALL of the literals on the branch, “positive” (the atomic sentences) and “negative” (the negated atomic sentences). [In practice, circle them.]
  - Step 2:** Set the FOL names that each literal employs as designating a number as follows: “a” designates 1, “b” designates 2, “c” designates 3, ...
  - Step 3:** For every freestanding identity sentence, (re-)set both the names to designate the lower of the two numbers involved from Step 2, and eliminate the higher number from consideration. So, if the identity sentence on the open branch is “ $(a = b)$ ”, then, after step 1 gives you that “a” designates 1 and “b” designates 2, you now reset “b” to designate 1 and eliminate 2 from consideration.
  - Step 4:** Put the number that each name from every literal on the branch designates into a set that is the domain of the interpretation, D.
  - Step 5:** Put the number designated by each name from every atomic (“positive”) literal on the open branch into the extension for the literal’s predicate (ordered pairs of numbers into the extension of binary predicates, ordered triples into the extension of trinary predicates, etc., to match the order that the names occur in the literal).
  - Step 6:** Check to make sure that, for each “negative” literal on the branch, the relevant items (individual numbers for one-place predicates, ordered pairs of numbers for binary predicates, ordered triples of numbers for trinary predicates, etc.) DO NOT OCCUR in the extension of the predicate employed in the literal.
  - Step 7:** Test the resulting *interpretation* on  $\Sigma$ , i.e., apply the interpretation to the sentences that are in  $\Sigma$  to see whether it makes them all true.
- That’s it.***

What you’ve constructed is an OBI. [Pronounced “Oh-Bee”, as in “Obi-Wan Kenobi”.]

So, consider the open tree constructed above. The positive literals on this branch are “ $F(a)$ ” and “ $G(a)$ ” (there are no negative literals on the branch). These sentences employ just one name, “a”, so set it to designate 1. That means the OBI is as follows.  $D = \{1\}$ , “F”:  $\{1\}$ , “G”:  $\{1\}$ . This interpretation makes “ $\forall x(F(x) \rightarrow G(x))$ ” true and makes “ $\exists x(F(x) \wedge G(x))$ ” true, so it shows that the set is **consistent**, as well as **compatible**.

**NOTE: The OBI method applies only to *finite* completed branches. The method does not apply to branches that are “open” due to infinite looping.**

To practice, use any of the practice problems or homework problems. If the tree is open, construct an OBI on (at least) one open branch.