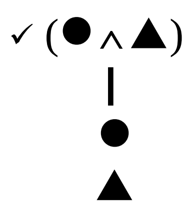
3.31. Analytic Trees: A Non-Semantic Test of Validity

For efficiency and convenience, truth trees serve as the high-water mark of our semantic tests of validity. But in this and the next section we explore tests of validity which, despite their striking resemblance to truth trees, make no reference to the semantic categories of “truth” and “falsehood”. These tests will offer deeper insight into the nature of the truth tree method, and its relation both to familiar semantic concepts and familiar families of sentences.

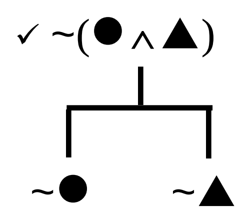
1. Analytic Trees. The first of these tests – the **analytic tree** method – introduces a seemingly modest change to truth trees.[1] Whereas in truth trees the only atoms – sentences not further broken down – are sentence letters, analytic trees treat **basics** – sentence letters and their negations – as atoms. This change opens the way to **eliminating truth and falsehood** from the tree test. For in place of the false sentence of truth trees (placed on the right of its line) we now have negations of those sentences; and un-negated sentences stand in for true sentences.

The True Conjunction and False Conjunction rules of truth trees, for instance, give way to rules for plain and Negated Conjunctions.

Conjunction



Negated Conjunction

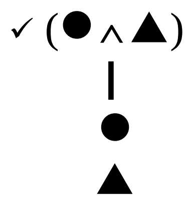


Two points bear comment. First, with truth and falsehood gone, gone as well is the need to put sentences on one or another side of the line. So every sentence will sit at the middle of its tree path. (In this respect, the current trees are similar to construction trees, which likewise make no reference to truth or falsehood.)

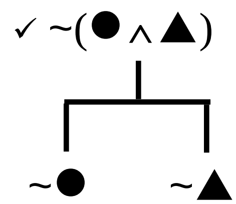
[1] In Smullyan 1968/1995: 20 these are called “unsigned analytic tableaux,” and truth trees are called “signed analytic tableaux”.

But in another respect analytic trees are unlike both truth trees and construction trees. For while truth tree rules break each sentence into its immediate construction part(s), that is not so here: the sentence “~(P ∧ Q)” is now analyzed into “~P” and “~Q,” neither of which is, construction-wise, a part of that negated conjunction.

Conjunction



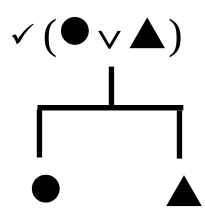
Negated Conjunction



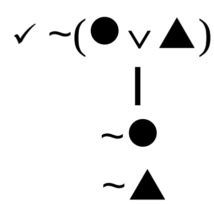
Still, these rules make sense semantically: whenever a conjunction is true, both of its parts are true. And when the negation of the conjunction is true, the negation of one or the other of the conjunction’s parts must likewise be true.

Similarly, if a disjunction is true, one or the other of its parts (or both) is as well. Whereas when the negation of the disjunction is true (making a “neither... nor” claim), so is the negation of both the disjunction’s parts.

Disjunction



Negated Disjunction



Finally, when a double negation of a sentence is true, that very sentence is as well.

Double Negation

✓ ~~●
|
●

Since a single-negation of a sentence letter is (as a basic) treated here as an atom, and so not analyzed further, **we have no need of a rule for single-negations**. So the analytic tree method requires fewer rules than the truth tree method.

Walking through a familiar valid example illustrates how analytic trees work.

1. We're either having ice cream or cake.
2. We're not having ice cream.

∴ We're having cake.

1. $(P \vee Q)$
2. $\sim P$

$\therefore Q$

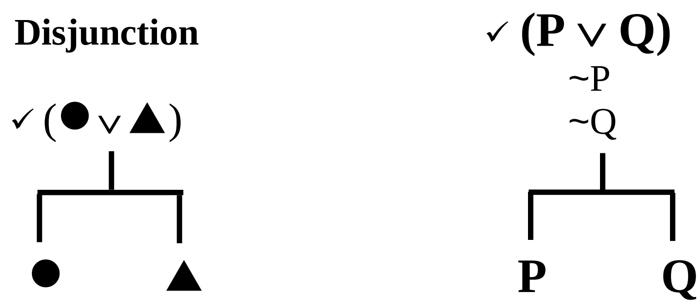
The truth tree test of validity pictured a validity counterexample for this argument by placing all the premises on the left, and the conclusion on the right. In analytic trees the validity counterexample is represented by the **counterexample set** for the argument: the **premises** and the **negation of the conclusion**.[2]

So analysis of our argument begins like so.

$(P \vee Q)$
$\sim P$
$\sim Q$

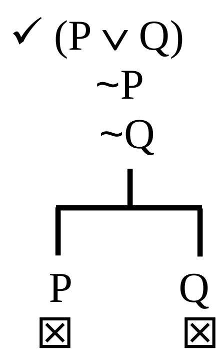
[2] Counterexample sets were introduced in "*3.20. Validity and Consistency*".

Treating negated sentence letters as atoms, the second and third sentences offer no occasion for analysis. That leaves only the first premise, “(P ∨ Q)”.



With all sentences larger than basics analyzed, we diagnose each path from bottom to top just as in truth tables – now looking only at the *basics* as we trace up each path. Instead of closing a path if a sentence letter is on both left and right sides, a path now closes if it contains both a **sentence letter** and also **its negation**.

Diagnosing in this way closes both paths – the left one because it contains both “P” and “~P,” the right because it contains both “Q” and “~Q”.

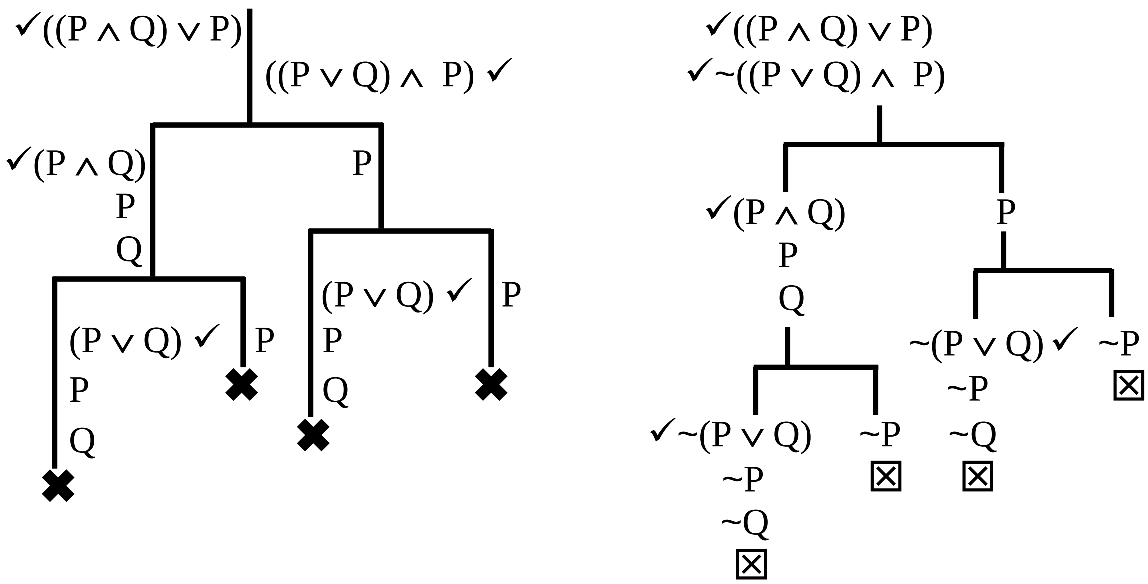


Thus the argument is judged valid.

The same heuristics and restrictions apply to analytic trees as did to truth trees: we break down non-branching sentences before branching ones, and hang a copy of a branch off of every open path below the branch-causing sentence.

What follows is a further example, with a truth tree of the same argument for comparison. The argument comes out valid on both tests.

$$\frac{1.\ ((P \wedge Q) \vee P)}{\therefore\ ((P \vee Q) \wedge P)}$$



But precisely because analytic trees seem like such a minor variation on truth trees, it bears stressing that all semantic categories have here been stripped away: simply through systematic disassembly of premises and conclusion, validity is assessed *with no appeal to truth or falsehood*. This non-semantic approach to validity is continued, in a quite different way, in the later discussion of deductions.[3]

[3] Beginning in "*3.35. Fundamentals of Deduction*".