Phil 57 section 3
Fall 2010

Truth-Tree Analysis

Truth-trees are a strategy for evaluating whether arguments are valid without having to construct a full truth-table.

Big idea: In a valid argument, if the premises are all true, the conclusion cannot be false. So, a diagram that tells you whether the conclusion can be false when the premises are true will tell you whether the argument is valid.

Why a truth-tree? Usually requires fewer rows than a truth-table (where 4 atomic statement letters requires 128 rows), and helps you break down complex formulae into atomic ones as you go.

How?
First, stack the premises and *the negation of the conclusion.*
Next, apply the following rules of truth-tree evaluation and construction.

Note on notation:
In the rules below, Δ and O stand for wffs (whether atomic or complex).
A checkmark (✓) placed next to a formula means it's been broken down.

Rules:

1. If in any connected branch of the tree a wff and its negation both occur, place an X underneath the formula to show that the branch is closed.

$$\begin{array}{l} O \\ \underline{\sim O} \\ X \end{array}$$

2. A wff with two negations applying directly to itself can be replaced on a line by the wff itself:

$$\begin{array}{l} \underline{\sim\sim O} \\ O \end{array}$$

3. Break down a negated conditional into a single trunk with the affirmation of the antecedent followed by the negation of the consequent:

$$\begin{array}{c} \underline{\checkmark \sim(O \rightarrow \Delta)} \\ O \\ \sim\Delta \end{array}$$

4. Break down a conditional into two separate trunks, one trunk with a negation of the antecedent and one with the affirmation of the consequent:

✓(O → Δ)
~O   Δ

5. Break down a conjunction into a single trunk with both conjuncts stacked:

✓(O • Δ)
O
Δ

6. Break down a disjunction into two trunks, one for each disjunct:

✓(O v Δ)
O   Δ

7. Break down a biconditional into two trunks, one that has both the antecedent and the consequent, the other that has the negation of both the antecedent and the consequent:

✓(O ↔ Δ)
O   ~O
Δ   ~Δ

8. Break down a negated conjunction into two trunks, each with the negation of one of the conjuncts:

✓~(O • Δ)
~O   ~Δ

9. Break down a negated disjunct into a single trunk with each disjunct negated:

✓~(O v Δ)
~O
~Δ

10. Break down a negated biconditional into two trunks, one with the negation of the antecedent and the affirmation of the consequent, the other with the affirmation of the antecedent and the negation of the consequent:

✓~(O ↔ Δ)
~O   O
Δ   ~Δ