Statements and Reduction Trees

Reduction trees can be used to examine statement forms and the relationships between statement forms. The procedure and rules are the same as with arguments, except the initial list consists of a single statement or statement form. The reduction tree rules are applied to the statement until either every path is closed or each unchecked line is either truth functionally simple or the denial of a truth functionally simple statement.

Contradictions

If a statement form is contradictory, then it has no possible true substitution instances. With no true substitution instances, every path of a reduction tree for the statement will close. This is illustrated by the following.

```
√  1. (p ⊃ q) & (p & ~q)        statement
√  2. p ⊃ q
√  3. p & ~q                    from 1, conjunction
   4. p
   5. ~q                        from 3, conjunction
        /      \
6. ~p           q               from 2, implication
    x           x
```

Tautologies

A tautology is a statement form with only true substitution instances. With no false substitution instances, a reduction tree for a tautology will have no paths close. This is illustrated in the tree below.

```
              √  1. (p ⊃ q) ⊃ [p ⊃ (p & q)]      statement
                     /              \
√  2. ~(p ⊃ q)            √ 2. p ⊃ (p & q)]      from 1, implication
   3. p    from 2,               /      \
   4. ~q   implication          /        \
      o
                         3. ~p       √ 3.p & q    from 3, implication
                            o           4. p
                                        5. q      from 3, conjunction
                                           o
```

This tree is complete with three open paths. We cannot infer, however, that the statement is a tautology from the above tree. As will be seen below, a contingent statement may have all paths open.

A denied tautology is a contradiction. Thus if we deny the above statement, every path should close.

√ 1. ~{(p ⊃ q) ⊃ [p ⊃ (p & q)]} — denied statement
√ 2. (p ⊃ q) — from 1, denied implication
√ 3. ~[p ⊃ (p & q)]
√ 4. p — from 3, denied implication
√ 5. ~(p & q)
6. ~p | q — from 2, implication
x (under ~p)
7. ~p | ~q (under q) — from 5, denied conjunction
x x

In the above tree, we see that each does. Thus we know the statement is a tautology.

Contingencies and Satisfiability

A contingent statement form has both true and false substitution instances. Since it has both, a reduction tree for either a contingent statement form or its denial should have open paths. Here is a tree for a contingent statement.

√ 1. (p ∨ q) ⊃ p — statement
√2. ~(p ∨ q) | p — from 1, implication
3. ~p | o
4. ~q — from 2, denied disjunction

Note that in the above tree, both paths remain open. We cannot infer from this, however, that the statement is a tautology. The denial of the above statement also has a open path.

√ 1. ~[(p ∨ q) ⊃ p] — denied statement
√ 2. p ∨ q
3. ~p — from 1, denied implication
4. p | q — from 2, disjunction
x o

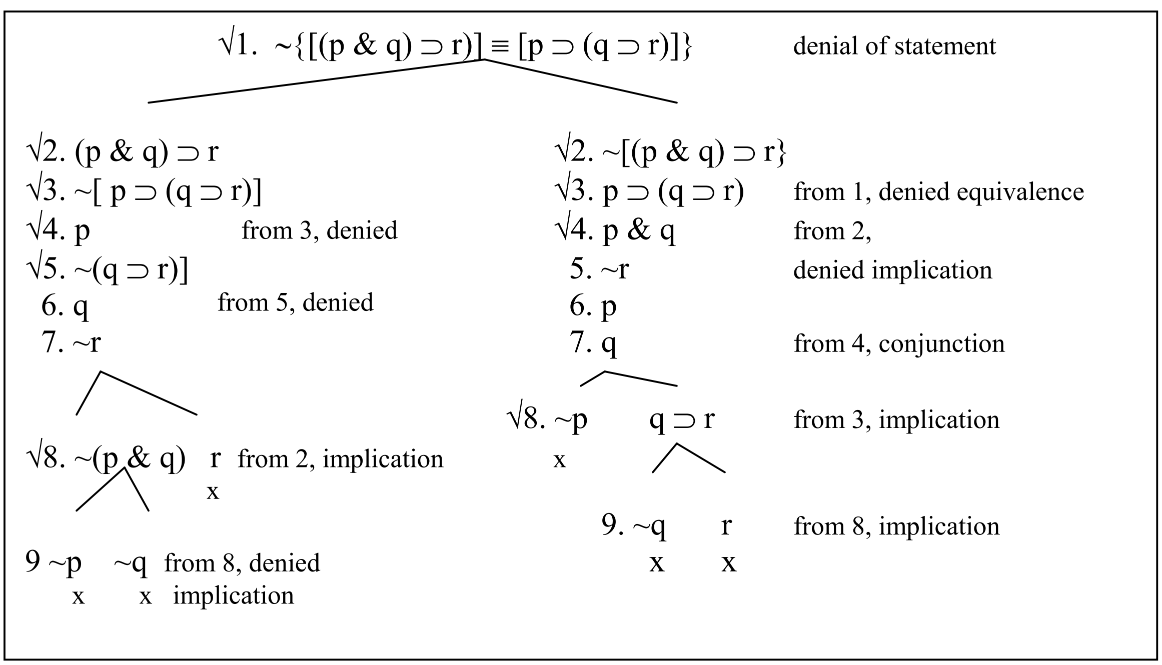
What a reduction tree reveals is whether or not a given statement form or set of statement forms is satisfiable. A statement form is satisfiable if and only if it has at least one true substitution instance. Since a reduction tree tests for satisfiability, and both contingencies and tautologies are satisfiable, the only way to prove a statement form is a tautology is to test its denial for unsatisfiability.

Logical Implication and Logical Equivalence

Reduction trees can be used to test for logical implication and logical equivalence in the same way that they can be used to test for tautologies. The statement form, $p \Rightarrow (p \vee q)$ is true if and only if $p \supset (p \vee q)$ is a tautology. Thus we test the truth of $p \Rightarrow (p \vee q)$ by testing whether $p \supset (p \vee q)$ is a tautology.

√	1.	$\sim[p \supset (p \vee q)]$	denial of statement
√	2.	p	from 1, denied implication
√	3.	$\sim(p \vee q)$	
	4.	$\sim p$	from 3, denied disjunction
	5.	$\sim q$	
		x	

Since the tree for the denial of $p \supset (p \vee q)$ closes, $p \Rightarrow (p \vee q)$ is true.
The statement form, $[(p \& q) \supset r)] \Leftrightarrow [p \supset (q \supset r)]$ is true if and only if $[(p \& q) \supset r)] \equiv [p \supset (q \supset r)]$ is a tautology. Thus the following tree.



Exercise 7.1

Use reduction trees to demonstrate whether the following statement forms are tautologous, contradictory, or contingent.

1. $p \equiv (p \vee q)$
2. $(p \vee q) \supset \sim(p \,\&\, q)$
3. $(p \equiv q) \supset \sim(p \,\&\, q)$
4. $p \supset (q \supset p)$
5. $\sim[(p \,\&\, q) \supset \sim(p \,\&\, \sim q)]$

Use reduction trees to demonstrate the following logical truths.

6. $[(p \supset q) \,\&\, p] \Rightarrow q$
7. $[(p \supset q) \,\&\, \sim q] \Rightarrow \sim p$
8. $[(p \supset q) \,\&\, (q \supset r)] \Rightarrow (p \supset r)$
9. $[(p \vee q) \,\&\, \sim q] \Rightarrow p$
10. $\{[(p \supset q) \,\&\, (r \supset s)] \,\&\, (p \vee r)\} \Rightarrow (q \vee s)$