

Mathematical Logic- Semantics (truth tables/ equivalent formulae)

August 8, 2022

The semantics of formulae built up syntactically using logical connectives is evaluated by iteratively applying the basic rules of semantics at each sub-formula (formula tree node).

The basic semantics are given in the following two tables.

Table 1: Negation/ NOT

p_1	$\neg p_1$
0	1
1	0

Table 2: Standard binry connectives

p_1	p_2	$p_1 \wedge p_2$	$p_1 \vee p_2$	$p_1 \Rightarrow p_2$	$p_1 \Leftrightarrow p_2$	$p_1 \oplus p_2$
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

Given an arbitrary syntactic formula in propositional logic, one can order the operators and then evaluate the formula under any assignment, by applying the basic rules shown in the preceeding two truth tabes.

Applying the following rules as appropriate, one can transform any foir-mula int one using only

- NOT and OR

- NOT and AND

Rules

1. $p \Rightarrow q \equiv \neg p \vee q$
2. $p \Leftrightarrow q \equiv ((p \Rightarrow q) \wedge (q \Rightarrow p))$
3. $p \oplus q \equiv \neg(p \Leftrightarrow q)$

The basic truth tables allow us to evaluate the semantics of any syntactic formula systematically. However, we now need to demonstrate the reverse problem:

Given any semantic formula in the form of a truth table, construct a syntactic formula for it using the standard logical connectives.

We will address this problem using two standard formats to represent any formula in propositional logic. These are:

1. **Conjunctive Normal Form (CNF)**: Here the formula is written as the AND of several clauses. A clause is the OR of several literals. a Literal is either a propositional variable or its negation.
2. **Disjunctive Normal Form (DNF)**: Here the formula is written as the OR of several clauses. A clause is the AND of several literals. a Literal is either a propositional variable or its negation.

We will use this to translate any formula from truth table form to CNF or DNF.

A **satisfying assignment** for a formula is any assignment that makes the formula evaluate to true. Computationally it is a difficult problem to know whether a formula has any satisfying assignments. The brute force way of trying every assignment is not ideal since a formula with k variables has 2^k assignments which is a large number. A formula that is satisfied under all possible assignments is called a **tautology** or **validity**. A formula that evaluates to false under all possible assignments is called a **contradiction**.

Two formulae are **(semantically) equivalent** if they evaluate to the same truth value under identical assignments. If ϕ and ψ are two formulae, then we say $\phi \Rightarrow \psi$ if **every** assignment that makes ϕ true, also make ψ true. This is consistent with our standard definition of the \Rightarrow logical connective. An example of such a pair of formula is: $\phi = p \wedge q$ and $\psi = p \vee q$. There are many others, and these can be used to make logical deductions. These are called **inference rules**.