Laws of Propositional Logic

Jayadev Misra

9/24/98

• (Commutativity and associativity) \( \land, \lor, \equiv \) are commutative and associative.

• (idempotence) \( \land, \lor \) are idempotent:
\[
p \lor p = p \\
p \land p = p
\]

• (Distributivity of \( \land \) and \( \lor \))
\[
p \lor (q \land r) = (p \lor q) \land (p \lor r) \\
p \land (q \lor r) = (p \land q) \lor (p \land r)
\]

• (absorption)
\[
p \land (p \lor q) = p \\
p \lor (p \land q) = p
\]

• (Laws with Constants)
\[
p \land \text{true} = p, \ p \land \text{false} = \text{false} \\
p \lor \text{true} = \text{true}, \ p \lor \text{false} = p \\
p \land \neg p = \text{false}, \ p \lor \neg p = \text{true} \\
p \equiv p \equiv \text{true}, \ p \equiv \neg p \equiv \text{false}
\]

• (Double Negation)
\[
\neg \neg p = p
\]

• (DeMorgan)
\[
\neg(p \land q) = (\neg p \lor \neg q) \\
\neg(p \lor q) = (\neg p \land \neg q)
\]

• (Implication)
\[
(p \Rightarrow q) = (\neg p \lor q) \\
(p \Rightarrow q) = (\neg q \Rightarrow \neg p)
\]

If \((p \Rightarrow q)\) and \((q \Rightarrow r)\) then \((p \Rightarrow r)\), i.e., \([p \Rightarrow q] \land (q \Rightarrow r)] \Rightarrow [p \Rightarrow r]

• (Equivalence)
\[
(p \equiv q) = (p \land q) \lor (\neg p \land \neg q) \\
(p \equiv q) = [(p \Rightarrow q) \land (q \Rightarrow p)]
\]

**Quantification** We use quantification in writing arithmetic and boolean expressions. In all cases, the form of a quantification is as follows: \((op \ dummy: range: body)\), where \(op\) is an operator, described below, \(dummy\) is a variable (or a list of variables), \(range\) defines the constraints on \(dummy\), and \(body\) is used to construct the expression. The \(op\) may be an arithmetic or boolean operator; it is required to be associative and commutative.
\(0 \leq i \leq N : A[i]\)
\(\forall i : 0 \leq i < N : A[i] \leq A[i + 1]\)
\(\forall i, j : 0 \leq i \leq N \land 0 \leq j \leq N \land i \neq j : M[i, j] = 0\)

To evaluate such an expression: (1) first compute all possible values of the dummy that satisfy range, (2) next instantiate the body with each value of the dummy computed in (1), and (3) finally, join the instantiated expressions in (2) using the operator op. Thus, the first expression is the sum of the values in array \(A[0..N]\). The second expression is true if \(A\) is sorted in ascending order.

The next expression has two dummies; it is a boolean expression that is true if all off-diagonal elements of matrix \(M[0..N, 0..N]\) are zero.

In quantified boolean expressions, we often use the existential quantifier, \(\exists\), and universal quantifier, \(\forall\), in place of \(\lor\) and \(\land\). We often omit the range to denote that the dummy ranges over all its possible values. Realize that

\((\forall i : q \land r : B)\) is same as \((\forall i : q : r \Rightarrow b)\), and
\((\exists i : q : b)\) is same as \((\exists i : q \land b)\).

**Exercise:** Show that
\((\forall i : q \land r : B)\) is same as \((\forall i : q : r \Rightarrow b)\), and
\((\exists i : q \land r : b)\) is same as \((\exists i : q : r \land b)\).

The following rules are extremely useful. In the first two cases \(i\) should not occur in \(p\).

\(p \lor (\forall i : q : b)\) is same as \((\forall i : q : p \lor b)\)
\(p \land (\exists i : q : b)\) is same as \((\exists i : q : p \land b)\)
(De Morgan)
\(\neg(\exists i : q : b)\) is same as \((\forall i : q : \neg b)\)
\(\neg(\forall i : q : b)\) is same as \((\exists i : q : \neg b)\)

**Exercise** Why are the following not valid?

\(p \land (\forall i : q : b)\) is same as \((\forall i : q : p \land b)\)
\(p \lor (\exists i : q : b)\) is same as \((\exists i : q : p \lor b)\)