Laws of Propositional Logic
Jayadev Misra
9/24/98

- (Commutativity and associativity) $\wedge, \vee, \equiv$ are commutative and associative.
- (idempotence) $\wedge, \vee$ are idempotent:
  $p \vee \ p = p$
  $p \wedge \ p = p$
- (Distributivity of $\wedge$ and $\vee$)
  $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$
  $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$
- (absorption)
  $p \wedge (p \vee q) = p$
  $p \vee (p \wedge q) = p$
- (Laws with Constants)
  $p \wedge \mathit{true} = p$, $p \wedge \mathit{false} = \mathit{false}$
  $p \vee \mathit{true} = \mathit{true}$, $p \vee \mathit{false} = p$
  $p \wedge \neg p = \mathit{false}$, $p \vee \neg p = \mathit{true}$
  $p \equiv p \equiv \mathit{true}$, $p \equiv \neg p \equiv \mathit{false}$
- (Double Negation)
  $\neg\neg p = p$
- (DeMorgan)
  $\neg(p \wedge q) = (\neg p \vee \neg q)$
  $\neg(p \vee q) = (\neg p \wedge \neg q)$
- (Implication)
  $(p \Rightarrow q) = (\neg p \vee q)$
  $(p \Rightarrow q) = (\neg q \Rightarrow \neg p)$
  If $(p \Rightarrow q)$ and $(q \Rightarrow r)$ then $(p \Rightarrow r)$, i.e., $[(p \Rightarrow q) \wedge (q \Rightarrow r)] \Rightarrow [p \Rightarrow r]$
- (Equivalence)
  $(p \equiv q) = (p \wedge q) \vee (\neg p \wedge \neg q)$
  $(p \equiv q) = [(p \Rightarrow q) \wedge (q \Rightarrow p)]$

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.

$(+i : 0 \le i \le N : A[i])$
$(\forall i : 0 \le i < N : A[i] \le A[i+1])$
$(\forall i, j : 0 \le i \le N \wedge 0 \le j \le N \wedge i \ne j : M[i,j] = 0)$

To evaluate such an expression: (1) first compute all possible values of the *dummy* that 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 $\vee$ and $\wedge$. We often omit the *range* to denote that the *dummy* ranges over all its possible values. Realize that

$(\forall i : q : b)$ is same as $(\forall i :: q \Rightarrow b)$, and
$(\exists i : q : b)$ is same as $(\exists i :: q \wedge b)$.

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

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

$p \vee (\forall i : q : b)$ is same as $(\forall i : q : p \vee b)$
$p \wedge (\exists i : q : b)$ is same as $(\exists i : q : p \wedge b)$
(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?

$p \wedge (\forall i : q : b)$ is same as $(\forall i : q : p \wedge b)$
$p \vee (\exists i : q : b)$ is same as $(\exists i : q : p \vee b)$