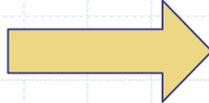


# Logic

March 28, 2005

# Propositional Logic

◆ Information definition: a **proposition** is a statement of **fact**

■ "It is raining" (english)  Raining

◆ Connectives: operators on propositions

■ And, or, not, implies, if and only if

$\wedge, \vee, \neg, \rightarrow, \leftrightarrow$

# Syntax

- ◆ Symbols:  $p, q, r, s, t$  (variables)
- ◆ Constants:  $T, F$
- ◆ Functions:  $f, g, h$  (n-ary) and connectives  
 $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$
- ◆ Relations:  $R, S$  (n-ary)
- ◆ Parentheses:  $), ($
- ◆ Equality  $\equiv$

# Examples

$$p \rightarrow q$$

$$(p \wedge \neg p) \vee r$$

$$)) \vee \wedge \equiv$$

# Formulas and Terms

## ◆ Rules:

- All symbols are formulas
- All constants are formulas
- If  $t_0, t_1$  are formulas then so are

$$t_0 \wedge t_1, t_0 \vee t_1, t_0 \rightarrow t_1, t_0 \leftrightarrow t_1, \neg t_0, (t_0)$$

- If  $t_0, t_1$  are formulas then so are

$$Rt_0t_1, t_0 \equiv t_1$$

- Formulas composed from symbols, constants and functions are called terms

# Semantics

$p$	$q$	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
$T$	$T$	$F$	$T$	$T$	$T$	$T$
$T$	$F$	$F$	$F$	$T$	$F$	$F$
$F$	$T$	$T$	$F$	$T$	$T$	$F$
$F$	$F$	$T$	$F$	$F$	$T$	$T$

# Semantics II

- ◆ Semantics of any formula is given by an evaluation function  $\Phi$  from formulas to  $\{T, F\}$
- ◆ To define the semantics, it suffices to define evaluation of symbols and functions (and use the previous slide)

# Examples II

- ◆ A tautology is always true  $p \vee \neg p$
- ◆ A contradiction is always false  $p \wedge \neg p$
- ◆ One way to derive truth of a formula is to use a truth table.

$$(p \rightarrow q) \equiv (\neg p \vee q)$$

# Laws of Propositional Logic

- ◆ Commutativity
- ◆ Associativity
- ◆ Distributivity
- ◆ DeMorgan

# Rules of Inference

- ◆ Modus Ponens
- ◆ Modus Tollens
- ◆ Syllogism
- ◆ Disjunctive Syllogism
- ◆ Specialization
- ◆ Conjunction

# Theories

- ◆ A **Theory** in propositional logic is a set of constants, functions, relations and axioms.
- ◆ Example: (theory of ordered integers)
  - Constants: non-negative integers
  - Function: +, Relation: <
  - Axioms:
    - $\neg(x < x)$
    - $0 < x \rightarrow y < x + y$
    - $(x < y) \rightarrow \neg(y < x)$

# Why?

- ◆ Why do computer scientists care?
- ◆ Because theories are *specifications* of a collection of structures
- ◆ To reason about code correctness
- ◆ To enable code transformations
  - Must preserve invariants