CSE 311 Foundations of Computing I

Autumn 2011, Lecture 3

Propositional Logic, Proofs, Predicate Calculus

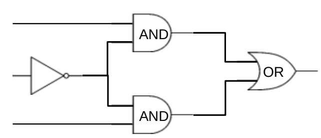
Administrative

- Course web: http://www.cs.washington.edu/311
  - Homework, Lecture slides, Office Hours ...
- Homework:
  - Due Wednesday at the start of class
- Paul Beame is travelling this week – so Richard Anderson will cover both lectures

Highlights of week 1

- Predicate calculus
- Basic logical connectives
- If pigs can whistle, then horses can fly

Combinational Logic Circuits



Design a 3 input circuit to compute the majority of 3. Output 1 if at least two inputs are 1, output 0 otherwise

What about a majority of 5 circuit?

Logical equivalence

- Terminology: A compound proposition is a
  - *Tautology* if it is always true
  - *Contradiction* if it is always false
  - *Contingency* if it can be either true or false

$p \vee \neg p$

$p \oplus p$

$(p \rightarrow q) \wedge p$

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

Logical Equivalence

- p and q are *logically equivalent* iff $p \leftrightarrow q$ is a tautology
  - *i.e.* p and q have the same truth table
- The notation $p \equiv q$ denotes p and q are logically equivalent
- Example: $p \equiv \neg\neg p$

p	$\neg p$	$\neg\neg p$	$p \leftrightarrow \neg\neg p$

De Morgan’s Laws

- $\neg (p \land q) \equiv \neg p \lor \neg q$
- $\neg (p \lor q) \equiv \neg p \land \neg q$

- What are the negations of:
  - The Yankees and the Phillies will play in the World Series
  - It will rain today or it will snow on New Year’s Day

De Morgan’s Laws

Example: $\neg (p \land q) \equiv (\neg p \lor \neg q)$

p	q	$\neg p$	$\neg q$	$\neg p \lor \neg q$	$p \land q$	$\neg(p \land q)$	$\neg(p \land q) \leftrightarrow (\neg p \lor \neg q)$
T	T						
T	F						
F	T						
F	F						

Law of Implication

Example: $(p \rightarrow q) \equiv (\neg p \lor q)$

p	q	$p \rightarrow q$	$\neg p$	$\neg p \lor q$	$(p \rightarrow q) \leftrightarrow (\neg p \lor q)$

Understanding connectives

- Reflect basic rules of reasoning and logic
- Allow manipulation of logical formulas
  - Simplification
  - Testing for equivalence
- Applications
  - Query optimization
  - Search optimization and caching
  - Artificial Intelligence
  - Program verification

Properties of logical connectives

- Identity
- Domination
- Idempotent
- Commutative
- Associative
- Distributive
- Absorption
- Negation

Equivalences relating to implication

- $p \rightarrow q \equiv \neg p \lor q$
- $p \rightarrow q \equiv \neg q \rightarrow \neg p$
- $p \lor q \equiv \neg p \rightarrow q$
- $p \land q \equiv \neg (p \rightarrow \neg q)$
- $p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$
- $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
- $p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$
- $\neg (p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Logical Proofs

- To show P is equivalent to Q
  - Apply a series of logical equivalences to subexpressions to convert P to Q
- To show P is a tautology
  - Apply a series of logical equivalences to subexpressions to convert P to **T**

Show $(p \wedge q) \rightarrow (p \vee q)$ is a tautology

Show $(p \rightarrow q) \rightarrow r$ and $p \rightarrow (q \rightarrow r)$ are not equivalent

Predicate Calculus

- *Predicate* or *Propositional Function*
  - A function that returns a truth value
- "*x* is a cat"
- "*x* is prime"
- "student *x* has taken course *y*"
- "$x > y$"
- "$x + y = z$"

Quantifiers

- $\forall x\, P(x)$: $P(x)$ is true for every x in the domain
- $\exists x\, P(x)$: There is an x in the domain for which $P(x)$ is true

Statements with quantifiers

Domain:
Positive Integers

Even(*x*)
Odd(*x*)
Prime(*x*)
Greater(*x*,*y*)
Equal(*x*,*y*)

- $\exists x$ Even(x)
- $\forall x$ Odd(x)
- $\forall x$ (Even(x) $\vee$ Odd(x))
- $\exists x$ (Even(x) $\wedge$ Odd(x))
- $\forall x$ Greater(x+1, x)
- $\exists x$ (Even(x) $\wedge$ Prime(x))

Statements with quantifiers

Domain:
Positive Integers

Even(x)
Odd(x)
Prime(x)
Greater(x,y)
Equal(x,y)

- $\forall\, x\, \exists\, y$ Greater (y, x)
- $\forall\, x\, \exists\, y$ Greater (x, y)
- $\forall\, x\, \exists\, y$ (Greater(y, x) $\wedge$ Prime(y))
- $\forall\, x$ (Prime(x) $\rightarrow$ (Equal(x, 2) $\vee$ Odd(x))
- $\exists\, x\, \exists\, y$(Equal($x$, y + 2) $\wedge$ Prime(x) $\wedge$ Prime(y))

Statements with quantifiers

Domain:
Positive Integers

Even(x)
Odd(x)
Prime(x)
Greater(x,y)
Equal(x,y)

- "There is an odd prime"
- "If x is greater than two, x is not an even prime"
- $\forall x \forall y \forall z$ ((Equal(z, x+y) $\wedge$ Odd(x) $\wedge$ Odd(y))$\rightarrow$ Even(z))
- "There exists an odd integer that is the sum of two primes"

Goldbach's Conjecture

- Every even integer greater than two can be expressed as the sum of two primes

Even(x)
Odd(x)
Prime(x)
Greater(x,y)
Equal(x,y)

Domain:
Positive Integers