CSE 311 Foundations of Computing I

Lecture 6

Predicate Logic

Autumn 2011

Announcements

- Reading assignments
  - Predicates and Quantifiers
    - 1.4, 1.5 7th Edition
    - 1.3, 1.4 5th and 6th Edition
- See clarification e-mail for HW 2 problem 4 (b)
  - $F(x, y, z) = x(yz + \bar{y}\,\bar{z})$

Highlights from last lecture

- Predicate Calculus
  - Predicate: A function that returns a truth value
  - 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
    - e.g. $\forall x\,(\text{Even}(x) \rightarrow \neg \text{Odd}(x))$
  - Multiple Quantifiers
    - $\forall x \exists y\, \text{Greater}(y, x)$

Notlargest(x)

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"
- $\forall x \forall y \forall z\, ((\text{Equal}(z, x+y) \wedge \text{Odd}(x) \wedge \text{Odd}(y)) \rightarrow \text{Even}(z))$
- "There exists an odd integer that is the sum of two primes"

English to Predicate Calculus

- "Red cats like tofu"

Cat(*x*)
Red(*x*)
LikesTofu(*x*)

$\forall\, x\, ((\text{Cat}(x) \wedge \text{Red}(x)) \rightarrow \text{LikesTofu}(x))$
$\forall\, x\, (\text{Cat}(x) \rightarrow (\text{Red}(x) \rightarrow \text{LikesTofu}(x)))$

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

Scope of Quantifiers

- Notlargest(x) $\equiv \exists$ *y* Greater (*y*, *x*)
  $\equiv \exists$ *z* Greater (*z*, *x*)
  - Value doesn't depend on y or z "bound variables"
  - Value does depend on x "free variable"
- Quantifiers only act on free variables of the formula they quantify
  - $\forall x (\exists y (P(x,y) \to \forall x\, Q(y, x)))$

Scope of Quantifiers

- $\exists x\ (P(x) \land Q(x))$ vs $\exists x\, P(x) \land \exists x\, Q(x)$

Nested Quantifiers

- Bound variable name doesn't matter
  - $\forall x \exists y\, P(x, y) \equiv \forall a \exists b\, P(a, b)$
- Positions of quantifiers can change
  - $\forall x (Q(x) \land \exists y\, P(x, y)) \equiv \forall x \exists y (Q(x) \land P(x, y))$
- BUT: Order is important...

Quantification with two variables

Expression	When true	When false
$\forall x \forall y\, P(x, y)$		
$\exists x \exists y\, P(x, y)$		
$\forall x \exists y\, P(x, y)$		
$\exists y \forall x\, P(x, y)$		

Negations of Quantifiers

- Not every positive integer is prime
- Some positive integer is not prime
- Prime numbers do not exist
- Every positive integer is not prime

De Morgan's Laws for Quantifiers

$$\neg \forall x\ P(x) \equiv \exists x\, \neg P(x)$$
$$\neg \exists x\, P(x) \equiv \forall x\, \neg P(x)$$

De Morgan's Laws for Quantifiers

$$\neg\forall x\ P(x) \equiv \exists x\ \neg P(x)$$
$$\neg\exists x\ P(x) \equiv \forall x\ \neg P(x)$$

"There is no largest integer"

$$\neg \exists x\ \forall y\ (x \geq y)$$
$$\equiv \forall x\ \neg \forall y\ (x \geq y)$$
$$\equiv \forall x\ \exists y\ \neg (x \geq y)$$
$$\equiv \forall x\ \exists y\ (y > x)$$

"For every integer there is a larger integer"

Logical Inference

- So far we've considered
  - How to understand and *express* things using propositional and predicate logic
  - How to *compute* using Boolean (propositional) logic
  - How to show that different ways of expressing or computing them are *equivalent* to each other
- Logic also has methods that let us *infer* implied properties from ones that we know
  - Equivalence is a small part of this

Applications of Logical Inference

- Software Engineering
  - Express desired properties of program as set of logical constraints
  - Use inference rules to show that program implies that those constraints are satisfied
- AI
  - Automated reasoning
- Algorithm design and analysis
  - e.g., Correctness, Loop invariants.
- Logic Programming, e.g. Prolog
  - Express desired outcome as set of constraints
  - Automatically apply logic inference to derive solution

Proofs

- Start with hypotheses and facts
- Use rules of inference to extend set of facts
- Result is proved when it is included in the set

An inference rule: Modus Ponens

- If p and p→q are both true then q must be true
- Write this rule as $\dfrac{p,\ p\rightarrow q}{\therefore\ q}$
- Given:
  - If it is Monday then you have a 311 class.
  - It is Monday.
- Therefore:
  - You have a 311 class