Harvard University
Computer Science 20

In-Class Problems 10

Wednesday, February 17, 2016

Author: Tom Silver

Executive Summary

1. Propositions and predicates
   - A proposition P is like a boolean variable. Its value is either "true" or "false."
   - A predicate $P(x)$ is like a boolean-valued function. It may have the value "true" for some values of the x and the value "false" for others.
   - A predicate can have more than one argument; e.g. Enrolled(x, y) might mean "student x is enrolled in course y."
   - In principle it is important to know the (possibly infinite) set of values D that x and y might assume (the domain of the function).
2. Quantifiers
   - The existential quantifier: $\exists x P(x)$ or $\exists x.P(x)$ or $\exists x$ s.t. $P(x)$ means "there exists at least one x in the domain D such that $P(x)$ is true.
   - The universal quantifier: $\forall x P(x)$ or $\forall x.P(x)$ means "for every x in the domain D, $P(x)$ is true.
3. Multiple quantifiers
   - $\exists x P(x)$ and $\forall x P(x)$ are both propositions, subject to the rules of logic that you already know.
   - $\exists x P(x, y)$ and $\forall x P(x, y)$ are both predicates of the form $Q(y)$, subject to the rules of quantificational logic that you are learning.
   - $\exists x \exists y P(x, y)$ and $\forall x \forall y P(x, y)$ are both propositions. The order of the quanitifiers is irrelevant.
   - $\exists x \forall y P(x, y)$ and $\forall y \exists x P(x, y)$ are both propositions, but they are different! The order of the quantifiers is important.
4. Negation and quantifiers
   - $\neg(\exists x.P(x)) \leftrightarrow \forall x.(\neg P(x))$.
   - $\neg(\forall x.P(x)) \leftrightarrow \exists x.(\neg P(x))$.

PROBLEM 1

Recall from Problem Set 1: Let A be the set of your pigeons, and let B be the set of pigeonholes in which they live. The *Generalized Pigeonhole Principle* states that for a natural number k, if $|A| > k|B|$, then there is a pigeonhole in which more than k pigeons live. Restate the GPP using quantifiers. Then negate the GPP (as you would for a proof by contradiction).

Solution.

PROBLEM 2

The domain of discourse is the set of all finite-length binary strings. The predicates Substring(x, y) (meaning x is a substring of y) and Prefix(x, y) (meaning x is a prefix of y) are available.

(A) Write an expression that means x consists of alternating 0s and 1s, e.g 01010 or 101010.
(B) Write an expression that means x is the binary representation of an integer of the form $5(2^k)$ for some integer $k \geq 0$.
(C) Write three different expressions that means x consists of one or more 1s, and no 0s. Important caveat: none of your expressions may contain "0".
(D) (BONUS) Write a fourth expression for (c) that is *recursive*, meaning that the expression appears in its own definition.

Solution.

PROBLEM 3

We define a committee to be a set of senators $\mathcal{S} = \{S_1, S_2, \cdots, S_n\}$. The predicate $M(S, \mathcal{C})$ means "Senator S is a member of committee $\mathcal{C}$." Rewrite the following in terms of predicate logic.

(A) Every committee has at least two senators serving on it.
(B) Any two distinct senators determine a unique committee.

Solution.

PROBLEM 4

The domain of discourse is the set of integers. Let $S(x, y, z)$ mean that "z is the sum of x and y."

(A) Write a formula that means x is an even integer.
(B) Write a formula that symbolizes the commutative property for addition $(x + y = y + x)$ of integers.
(C) Write a formula that symbolizes the associative law for addition
$(x + (y + z) = (x + y) + z)$ of integers.

Solution.