Solutions to In-Class Problems Week 3, Wed.

Problem 1.
For each of the logical formulas, indicate whether or not it is true when the domain of discourse is $\mathbb{N}$ (the nonnegative integers 0, 1, 2, ...), $\mathbb{Z}$ (the integers), $\mathbb{Q}$ (the rationals), $\mathbb{R}$ (the real numbers), and $\mathbb{C}$ (the complex numbers).

$$\begin{array}{rrrcl}
 & \exists x & (x^2 & = & 2) \\
\forall x & \exists y & (x^2 & = & y) \\
\forall y & \exists x & (x^2 & = & y) \\
\forall x \neq 0 & \exists y & (xy & = & 1) \\
\exists x & \exists y & (x+2y & = & 2) \wedge (2x+4y=5)
\end{array}$$

Solution.

Statement	$\mathbb{N}$	$\mathbb{Z}$	$\mathbb{Q}$	$\mathbb{R}$	$\mathbb{C}$
$\exists x\ (x^2 = 2)$	F	F	F	$T\ (x = \sqrt{2})$	T
$\forall x \exists y\ (x^2 = y)$	T	T	T	$T\ (y = x^2)$	t
$\forall y \exists x\ (x^2 = y)$	F	F	F	F (take $y < 0$)	t
$\forall x \neq 0 \exists y\ (xy = 1)$	F	F	T	$T\ (y = 1/x)$	T
$\exists x \exists y\ (x+2y=2) \wedge (2x+4y=5)$	F	F	F	F	F

■

Problem 2.
The goal of this problem is to translate some assertions about binary strings into logic notation. The domain of discourse is the set of all finite-length binary strings: λ, 0, 1, 00, 01, 10, 11, 000, 001, ... (Here λ denotes the empty string.) In your translations, you may use all the ordinary logic symbols (including =), variables, and the binary symbols `0`, `1` denoting 0, 1.

A string like $\texttt{01}x\texttt{0}y$ of binary symbols and variables denotes the *concatenation* of the symbols and the binary strings represented by the variables. For example, if the value of x is `011` and the value of y is `1111`, then the value of $\texttt{01}x\texttt{0}y$ is the binary string `0101101111`.

Here are some examples of formulas and their English translations. Names for these predicates are listed in the third column so that you can reuse them in your solutions (as we do in the definition of the predicate NO-1S below).

Meaning	Formula	Name
x is a prefix of y	$\exists z\ (xz = y)$	PREFIX(x, y)
x is a substring of y	$\exists u \exists v\ (uxv = y)$	SUBSTRING(x, y)
x is empty or a string of 0's	NOT(SUBSTRING$(\texttt{1}, x)$)	NO-1S(x)

(a) x consists of three copies of some string.

Solution. $\exists y\ (x = yyy)$ ■

(b) x is an even-length string of 0's.

Solution. $\text{NO-1S}(x) \wedge \exists y\ (x = yy)$ ■

(c) x does not contain both a 0 and a 1.

Solution. $\text{NOT}[\text{SUBSTRING}(\mathtt{0}, x) \text{ AND } \text{SUBSTRING}(\mathtt{1}, x)]$ ■

(d) x is the binary representation of $2^k + 1$ for some integer $k \geq 0$.

Solution. $(x = \mathtt{10}) \text{ OR } (\exists y\ (x = \mathtt{1}y\mathtt{1} \text{ AND } \text{NO-1S}(y)))$ ■

(e) An elegant, slightly trickier way to define $\text{NO-1S}(x)$ is:

$$\text{PREFIX}(x, \mathtt{0}x). \qquad (*)$$

Explain why (*) is true only when x is a string of 0's.

Solution. Prefixing x with 0 rightshifts all the bits. So the nth symbol of x shifts into the $(n+1)$st symbol of $0x$. Now for x to be a prefix of $0x$, the $n+1$st symbol of $0x$ must match the $(n+1)$st symbol of x. So if x satisfies (*), the nth and $(n+1)$st symbols of x must match. This holds for all $n > 0$ up to the length of x, that is, *all* the symbols of x must be the same. In addition, if $x \neq \lambda$, it must start with 0. Therefore, if x satisfies (*), all its symbols must be 0's.

Note that it's easy to see, conversely, that if $x = \lambda$ or x is all 0's, then of course it satisfies (*). ■

6.042J / 18.062J Mathematics for Computer Science
Spring 2010