CS6110/6116 Friday Feb 10, 2012

CS6110 Lect 9
CS6116 Lect 3

Mathematical and Philosophical Preliminaries - part 2

Our subject is programming languages, and as we have seen, it can be used to introduce the concept of a computable function, and somewhat unavoidably, some notion of data type, e.g. λ-terms, natural numbers, lists, pairs.

We have seen basically two control structures, case and recursion, but they can be encoded into our rich data type λ-terms.

So far, only the formalism of primitive recursion (PRA) has given us a handle on termination. This is critical to expressing mathematical ideas. In this lecture we examine more closely how Primitive Recursive Arithmetic (of T. Skolem, circa 1923) can be used to express mathematics. This was Kurt Gödel's "PL of choice" - not much to choose from in 1931.

CS6110/6116 Friday Feb. 10, 2012

Expressing mathematics in a programming language

We use add, mult and these functions to define logic.

$\mathrm{pred}(0) = 0$ $\qquad$ $\mathrm{monus}(x,0) = x$

$\mathrm{pred}(S(x)) = x$ $\qquad$ $\mathrm{monus}(x, S(y)) = \mathrm{pred}(\mathrm{monus}(x,y))$

Let $x \dot{-} y$ abbreviate $\mathrm{monus}(x,y)$.

Define the positive difference $|x,y|$ by

$$|x,y| = (x \dot{-} y) + (y \dot{-} x)$$

Define $a = b$ for a and b numbers to be a true proposition iff $|a,b| = 0$.

Goodstein defines $\alpha(|a,b|)$ as the number of the proposition $a = b$.

We can define the logical operators on propositions as follows. Let p, q be propositions $a=b$, $c=d$ respectively. Then

$\sim p$ is $(1 \dot{-} |a,b|) = 0$

$p \& q$ is $(|a,b| + |c,d|) = 0$

$p \vee q$ is $(|a,b| \cdot |c,d|) = 0$

$p \to q$ is $\sim p \vee q$

$p \leftrightarrow q$ is $(p \to q) \& (q \to p)$

Goodstein defines these "quantifiers."

$A_x^n(f(x) = 0)$ for all x from 0 to n $f(x) = 0$

$E_x^n(f(x) = 0)$ for some x from 0 to n $f(x) = 0$

$L_x^n(f(x) = 0)$ the least x from 0 to n $f(x) = 0$

We can validate mathematical induction

$$[P(0) \,\&\, A_x^n(P(x) \to P(x+1))] \to P(n)$$