Reiknirit, rökfræði og reiknanleiki

Magni Þór Birgisson

skil 5

1 Exercise 5.1 bls 126

write completely format definitions of the following functions:

1) the levell function lev(x),
which return 0 if x equals 0 and returns 1 otherwise;

$$\begin{aligned} \text{lev}(0) &= 0 \\ \text{lev}(i+1) &= succ(zero(P^2_1(i, \text{lev}(i)))) \end{aligned}$$

2) its complement is_zero(x)

$$\begin{aligned} \text{is\_zero}(0) &= 1 \\ \text{is\_zero}(i+1) &= zero(P^2_1(i, \text{is\_zero}(i))) \end{aligned}$$

3) the function of two arguments minus(x,y),
which returns $x - y$ (or 0 whenever $y \geq x$);

$$\begin{aligned} \text{dec}(0) &= 0 \\ \text{dec}(i+1) &= P^2_1(i, \text{dec}(i)) \end{aligned}$$

$$\begin{aligned} \text{minus}(x, 0) &= P^1_1(x) \\ \text{minus}(x, i+1) &= \text{dec}(P^3_2(i, \text{minus}(x, i), x)) \end{aligned}$$

4) the function of two arguments mult(x,y),
which returns the product og x and y;

$$\begin{aligned} \text{add}(0) &= P^1_1(x) \\ \text{add}(i+1) &= succ(P^3_2(i, \text{add}(i, x), x)) \end{aligned}$$

$$\begin{aligned} \text{mult}(0, y) &= 0 \\ \text{mult}(1, y) &= P^1_1(x) \\ \text{mult}(i+1, y) &= \text{add}(y, P^3_2(i, \text{mult}(i, y), y)) \end{aligned}$$

5) the "guard" function $x \# y$,
which returns o if x equals 0 and returns y otherwise
(verfy that it can be defined so as to avoid evaluting y
whenever x equals 0).

$$\begin{aligned} \text{guard}(0, y) &= 0 \\ \text{guard}(i+1, y) &= P^3_3(i, \text{guard}(i, y), y)) \end{aligned}$$

2 Exercise 5.3 bls 128

Verify that definition by cases is primitive recursive. That is, given primitive recursive function g and h and primitive recursive predicate P,
the new function f definded by

$$f(x1,\ldots,x_n) = \begin{cases} g(x_1,\ldots,x_n) & \text{if } P(x_1,\ldots,x_n) \\ h(x_1,\ldots,x_n) & \text{otherwise} \end{cases}$$

is also primitive recursive. (We can easily generalize this definition to multiple disjonit predicates defining multiple cases.) Further verify that this definition can be made so as to avoid evalution of the function(s) specified for the case(s) ruled out by the predicate.

$$f(x_1,\ldots,x_n) = \text{con}_2(P(x_1,\ldots,x_n)\#g(x_1,\ldots,x_n), \text{not}(P(x_1,\ldots,x_n)\#h(x_1,\ldots,x_n)))$$

3 Exercise 5.5 bls 129

Using the carious constructor of the last few exercises, prove that the following predicates and functions ar primitve recursive:

- $f(x, z_1, \ldots, z_n) = \text{min } y \;\; \leq x[P(y, z_1, \ldots, z_b)]$ returns the smallest y no larger then x such that the predicate P is true; if no such y exists, the function returns $x+1$.

  $$\begin{cases} f(n, z_1, \ldots, z_n) & \text{ef } f(n, z_1, \ldots, z_n) \leq n \\ n+1 & \text{ef } P(n+1, z_1, \ldots, z_n) \\ n+2 & \text{annars} \end{cases}$$

- $x \leq y$, treu if and only if x is no larger the y.

  is_zero(minus(x, y))

- $x|y$, true if and only if x divides y exactly.

  $\exists z < y|\text{mult}(x, z) = y$
  Einhver z sem er minni en y þar sem mult(x,z) = y

- is_prime(x), true if and only if x is prime.

  $\forall y < x$ and $(1 > y)|\text{not}(y|x)$
  fyrir öll y sem eru minni en x og stærri en 1 þá ekki hægt að deila þeim saman.

- prime(x) return the xth prime.

  Hérna þrufum við að prófa allar tölur prime(z) þangað til að við erum kominn með réttan fjölda og þá skilum við henni

4 Exercise 5.8 bls 136

We remarked earlier that any attempted enumeration of total functions, say $\{f_1, f_2, \dots\}$ is subject to diagonalization and tus incomplete, since we can always define the new total function $g(n) = f_n(n) + 1$ that does not apper in the enumeration. Thus the total functions cannot be enumerated. Why dows this line of reasoning not apply directly to the recursive functions?

Recursive föll $\leq$ Turingvél.
Við getum codað Turingvél með $\sum = \{0, 1\}$ og því er hún númeranleg.
Þar af leiðandi er Recursive föll það líka.