MA 3219 – Computability Theory

Frank Stephan. Departments of Computer Science and Mathematics, National University of Singapore, 3 Science Drive 2, Singapore 117543, Republic of Singapore

Email fstephan@comp.nus.edu.sg
Webpage `http://www.comp.nus.edu.sg/~fstephan/computability.html`
Telephone office +65-6874-7354
Office room number S14#06-06
Office hours Thursday 15.00-17.00h

Assignment for 02.02.2005. Can be corrected on request, it is not obligatory to hand the homework in.

1. Recursion. Determine the functions h which are defined as

$$h(x,y) = \begin{cases} f(x) & \text{if } y = 0; \\ g(x, y-1, h(x, y-1)) & \text{if } y \geq 1; \end{cases}$$

given f, g as follows:

(a) $f(x) = x$, $g(x,y,z) = z + 1$;
(b) $f(x) = 2^x$, $g(x,y,z) = 3 \cdot z$;
(c) $f(x) = x$, $g(x,y,z) = x + y + z$;
(d) $f(x) = 1$, $g(x,y,z) = (y+1) \cdot z$;
(e) $f(x) = 1$, $g(x,y,z) = 0$.

2. URM Programming. Write an URM program with three inputs r_1, r_2, r_3 such that the output is 0 if one of the inputs is 0 and the output is the least common multiple of three inputs r_1, r_2, r_3 if those are all greater than 0. You can write the program as a flowchart.

3. Bounded Minimalization. Given natural numbers $a_0, a_1, a_2, a_3, a_4, b_0, b_1, b_2, b_3, b_4$, show that there is a computable function doing the following:

- if $a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4$ then for some natural number x then it should output $x + 1$ for the least such x.
- if $a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \neq b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4$ for all natural numbers x then it should output 0.

For doing this, find an upper bound c (depending on $a_0, a_1, a_2, a_3, a_4, b_0, b_1, b_2, b_3, b_4$) such that an equality occurs below c whenever there is an equality at all and then search for the equality with bounded minimalization. You can use any of the functions on pages 36 and 37 of Cutland's book.

4. Minimalization. Given a number x, define $g(x)$ to be $x/2$ if x is even and $3x+1$ if x is odd. Show that there is a computable function u which counts the number of updates of the form $x = g(x)$ which are necessary until the value reached is 1. Note that $u(1) = 0$; $u(2) = 1$ since $g(2) = 1$; $u(3) = 7$ since $g(3) = 10$, $g(g(3)) = 5$, $g(g(g(3))) = 16$, $g(g(g(g(3)))) = 8$, $g(g(g(g(g(3))))) = 4$, $g(g(g(g(g(g(3)))))) = 2$ and $g(g(g(g(g(g(g(3))))))) = 1$. Furthermore, $u(0)$ is undefined since $g(0) = 0$. Prove that u is computable by using first recursion and then minimalization.