CS 590 Theory of Computation: Spring 1999
Homework 0

Due 25 January 1999 (Lecture 6)

Problems from the text are denoted by C or BC, depending on whether they are from the Cutland text or from the Bovet and Crescenzi text, followed by the chapter and problem number, separated by a period. For example, C 2.3 3 is problem 3 for chapter 2, section 3 of Cutland.
Question 1 (C 1.3.3 a,b,c pg. 21) Show that the following functions are computable by devising programs that will compute them:

(a) \( f(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases} \)

(b) \( f(x) = 5 \)

(c) \( f(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases} \)

Answer

(a) \( f(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases} \)

This program uses our convention that \( r_2 = 0 \) for this 1-ary function \( f \).

\[
P = \begin{cases} 
J(1, 2, 0), & \text{if } r_1 = r_2 = 0 \text{ then halt} \\
Z(1), S(1) & \text{else set } r_1 = 0 + 1 = 1 
\end{cases}
\]

(b) \( f(x) = 5 \)

\[
P = (Z(1), S(1), S(1), S(1), S(1), S(1))
\]

Notice that the program would be incorrect without the \( Z(1) \), since without that it would compute \( f(x) = x + 5 \).

(c) \( f(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases} \)

This program stores \( x \) into \( R_3 \), compares \( x \) and \( y \) (now in \( R_3 \) and \( R_2 \)), then returns the appropriate value. The statement numbers are listed only for convenience.

\[
P = \begin{cases} 
T(1, 3), & \text{save } r_1 \text{ in } R_3 \\
Z(1), & \text{set up to return zero} \\
J(3, 2, 0), & \text{if } x = r_3 = r_2 \text{ then return zero} \\
S(3) & \text{else return one} 
\end{cases}
\]
Question 2 (C 1.3.3 pg. 22) Suppose that \( P \) is a program without any jump instructions. Show that there is a number \( m \) such that either \( f^{(1)}_{m}(x) = m \) for all \( x \), or \( f^{(1)}_{m}(x) = x + m \) for all \( x \).

Answer Since \( f \) is a 1-ary function, we may assume that \( r_2, r_3, \ldots \) are all equal to zero.

Since there are no jump instructions, the instructions in \( P \) are executed once, in order. Also, let \( P = (I_1 \cdots I_n) \) (that is, assume \( P \) has \( n \) instructions).

Let \( k \) be the greatest instruction index such that \( I_k \) is either \( S(1) \) or \( T(m, 1) \) for some \( m > 1 \), and \( k = 0 \) if there are no such instructions (that is, the \( k \)th instruction is the last that could possible set \( R_1 \) to zero). Further, suppose there are \( m \geq 0 \) \( S(1) \) instructions in \( I_{k+1} \cdots I_n \).

If \( k > 0 \) then \( I_k \) resets \( r_1 \) to zero, and the following \( m \) \( S(1) \) instructions increment it \( m \) times, so that the program will return \( m \) for all inputs.

On the other hand, if \( k = 0 \), then the input value \( x \) is incremented \( m \) times in \( P \), so the program returns \( x + m \) for all inputs.
Question 3 (C 1.4.3 a,b,c pg. 23) Show that the following predicates are decidable.

(a) “$x < y$”

(b) “$x \neq 3$”

(c) “$x$ is even”

Answer

For each predicate $M$, we construct a program to compute $C_M$.

(a) “$x < y$”

This program searches for the value of the lesser of $x$ and $y$, and returns a one if that value equals $x$ and a zero otherwise. It uses $R_3$ as a counter and $R_4$ (initially zero) as the return value.

(b) “$x \neq 3$”

This program sets $r_2$ to three, then compares this to the input and returns the appropriate value. We use the convention that $r_k = 0$ for all $k > \rho(P) = 2$ (we only use registers $R_1$ and $R_2$).

(c) “$x$ is even”

This program scans for the value of $x$ and returns the appropriate value.
Question 4  Show that the predicates in the last question are computable.

Answer

You can’t. Predicates are *decidable*, only functions are *computable*.