Due: Friday, November 20, beginning of tutorial

NOTE: Each problem set counts 15% of your mark, and it is important to do your own work. You may consult with others concerning the general approach for solving problems on assignments, but you must write up all solutions entirely on your own. Copying assignments is a serious academic offence and will be dealt with accordingly.

1. Write a register machine program which computes the function $f(x) = 2x$. Be sure to respect our input/output conventions for register machines. Write your program in the style of the Copy program given on page 55 of the Notes. DO NOT USE MACROS. (Write your program in the original RM machine language.) Explain your program so that the marker can easily understand it.

2. Show that the following two problems are primitive recursive, using results from the Notes.

   a) $Bit(x, i) =$ the coefficient of 2^i in the binary notation of x. For example, the binary notation for 6 is 110, so $Bit(6, 0) = 0$, $Bit(6, 1) = Bit(6, 2) = 1$, and $Bit(6, i) = 0$ for $i > 2$.

   b) $NumOnes(x) =$ the number of 1's in the binary notation for x. For example, $NumOnes(6) = 2$.

3. Let PRIMES be the set of prime numbers.

   Let $A = \{x \mid \text{dom}(\{x\}_1) \subseteq \text{PRIMES}\}$.
   Let $B = \{x \mid \{x\}_1 \text{ is a primitive recursive function}\}$.

   Which of $A, \overline{A}, B, \overline{B}$ is recursive? Which is r.e.? Justify your answers. You may use the S-m-n Theorem (page 74) and the Kleene Normal Form Theorem (page 68). Or you may avoid using the S-m-n Theorem, and use Church's Thesis instead. (Do not use Rice's Theorem.)

4. The class $\mathcal{E}$ of Kalmar elementary functions can be defined as follows:

   We say that f is defined from g, h, and B by *limited recursion* iff

   $f(\vec{x}, 0) = g(\vec{x})$
   $f(\vec{x}, y + 1) = \min\{h(\vec{x}, y, f(\vec{x}, y)), B(\vec{x}, y)\}$

   Let $INIT(\mathcal{E}) = \{Z, S\} \cup \{I_{n,i}, 1 \leq i \leq n\} \cup \{f_+, E\}$ where $Z, S, I_{n,i}$ are the Initial Functions defined on page 57 of the Notes, and $f_+(x, y) = x + y$, and $E(x) = 2^x$.

   Definition: $f \in \mathcal{E}$ iff f can be obtained from $INIT(\mathcal{E})$ by finitely many applications of composition and limited recursion.

   Define the sequence of functions $E_0, E_1, ...$ by $E_0(x) = x$ and $E_{k+1}(x) = 2^{E_k(x)}$ for $k \geq 0$. Thus $E_k(x)$ is a superexponential function represented by a stack of k 2's with an x at the top. Note that $E_k \in \mathcal{E}$ for each k.

For an RM program $\mathcal{P}$ let $Time_\mathcal{P}(\vec{x})$ be the number of steps taken by $\mathcal{P}$ before $\mathcal{P}$ halts on input $\vec{x}$, or $Time_\mathcal{P}(\vec{x}) = \infty$ if $\mathcal{P}$ does not halt on input $\vec{x}$.

Definition: Let $\mathcal{C}$ be the class of all functions f such that there exists an RM program $\mathcal{P}$ and $k \in \mathbb{N}$ such that $\mathcal{P}$ computes f and $Time_\mathcal{P}(\vec{x}) \leq E_k(x_1 + ... + x_n)$ for all $\vec{x}$.

Theorem: (Ritchie-Cobham) $\mathcal{E} = \mathcal{C}$.

You are to prove the direction $\mathcal{C} \subseteq \mathcal{E}$. We start by observing that every specific function proved to be primitive recursive in the Notes *Computability Theory* is in fact in $\mathcal{E}$. This is because every application of primitive recursion can be replaced by limited recursion, since each such function $f(\vec{x}, y)$ is bounded above by $E_k(x_1 + ... + x_n + y)$ for some k, so we can take $B(\vec{x}, y) = E_k(x_1 + ... + x_n + y)$.

(a) Let $\mathcal{P}$ be an RM program, and let $STATE_\mathcal{P}(\vec{x}, t)$ be the number encoding the state of $\mathcal{P}$ after t steps of computation, where the initial state is $u_0 = p_0^0 p_1^{x_1} ... p_n^{x_n}$ (i.e. $x_1, ..., x_n$ are stored in registers $R_1, ..., R_n$, with program counter $K = 0$, as described on the bottom of page 58). Prove that $STATE_\mathcal{P} \in \mathcal{E}$. You may assume that the function $Nex(u, z)$ defined on page 67 is in $\mathcal{E}$.

(b) Now use (a) to show $\mathcal{C} \subseteq \mathcal{E}$.