Higher Set Theory - Classical and Ordinal Computability

Exercise Sheet 1
due on Tuesday, 12 April 2011

1. a) Let $\mathbb{A} = \{a_0, a_1, \ldots a_{k-1}\}$ be a finite alphabet. Write down explicitly an n-register program P over $\mathbb{A}$ (for a suitable n) that determines the shortest of two words over $\mathbb{A}$: For any $v, w \in \mathbb{A}^*$ a computation by P on an initial register content (*input*) $R(0)$, where $R(0)_1 = v$ and $R(0)_2 = w$ and all other registers are empty should halt and output a_0 if v is shorter (in the obvious sense) than w, and a_1 otherwise.

   b) Let $\mathbb{A} = \{a, b\}$. Write down explicitly an n-register program P over $\mathbb{A}$ (for a suitable n) that copies the content of one register to another: For any $w \in \mathbb{A}^*$ a computation by P on input $R(0)$, where $R(0)_1 = v$ and all other registers are empty should halt and output w Can one modify the program to either have $R(\theta)_1 = v$ or $R(\beta)_1 = \Box$ at halting time θ?

(4 points)

2. Now consider $\tilde{\mathbb{A}}$ for $\mathbb{A} = \{a, b\}$.

   (a) Write down explicitly an n-register program P over $\tilde{\mathbb{A}}$ (for a suitable n) that recognizes numerals:
   Input: $R(0)$ where $R(0)_1 = w$ for some word $w \in \tilde{\mathbb{A}}^*$ and all other registers empty
   Output: a if w is a numeral, b otherwise

   (b) Write down explicitly an n-register program P over $\tilde{\mathbb{A}}$ (for a suitable n) that recognizes register-commands:
   Input: $R(0)$ where $R(0)_1 = w$ for some word $w \in \tilde{\mathbb{A}}^*$ and all other registers empty
   Output: a if w is a register-command, b otherwise

   (c*) *(optional, no points awarded)* Give a sketch for a register program P over $\tilde{\mathbb{A}}$ (for a suitable n) that recognizes register programs.

(4 points)

3. In 1989, Leonore Blum, Michael Shub, and Stephen Smale introduced a kind of register machines that handles computations on arbitrary rings and fields. Today, these machines, called *Blum-Shub-Smale (BSS) machines*, are a widely used model for computation on real numbers. We give the following definition of *n-register-BSS machines* somewhat resembling our definition of register machines:

   Let R be a ring. Consider the alphabet $\mathbb{A} = R \cup \{g : R^n \to R^n \mid g \text{ is polynomial}\} \cup \{h : R^n \to R \mid h \text{ is polynomial}\} \cup \omega \cup \{x, :, =,), (, \to, h, a, l, t, ;\}$. An *n-register-BSS command* over R is a word over $\mathbb{A}$ of the following form:

- $x := g(x)$, where $g : R^n \to R^n$ is polynomial (replace the entire register content $x \in R^n$ by $g(x) \in R^n$)
- $h(x) := 0 \to k/l$, where $h : R^n \to R$ is polynomial, $k, l \in \omega$ (if $h(x) = 0$ where $x \in R^n$ is the entire register content, then jump to k, else jump to l).
- *halt* (output x_0)

  An *n-register-BSS program* over R is a word over $\mathbb{A}$ of the form

  $$c_0; c_1; \ldots; c_m - 1; halt$$

  where

  - each c_i is a n-register-BSS command
  - if c_i ends in $\to k/l$ then $k, l < m$

  Give a definition of *BSS-computations* by a program P.

(4 points)