Martin Otto
Stephane Le Roux

Exercises No.9

Exercise 1 Write R-programs over the binary alphabet $\mathbb{B} = \{0, 1\}$ for the following.

(a) A reverse copy program, which maps the reverse u^{-1} of the content u of R_1 into R_2 (assuming that all registers but R_1 are initially empty). More precisely, we want P_1 on $\overline{u} = (u, \Box, \Box, \ldots)$ to terminate with register contents $(u, u^{-1}, \Box, \ldots)$. Note that u is meant to be kept in R_1. How do we get a copy subroutine "$R_2 := R_1$" from this?

(b) An equality checking program P_2 which decides the condition $R_1 = R_2$ in the sense that P_2 terminates on any input $\overline{u} = (u_1, u_2, \Box, \ldots)$ and terminates with $R_3 = \Box$ iff $u_1 \neq u_2$.

(c) Combine the elements of these programs to obtain a program that decides the set $\textsc{Palindrome} = \{w \in \mathbb{B}^* : w = w^{-1}\}$.

Exercise 2 Write an R-program over the binary alpbabet $\mathbb{B} = \{0, 1\}$ that computes the successor function in terms of binary representations of natural numbers. (For definiteness: use binary representations with least significant bit to the right, so that $u = b_{n-1} \ldots b_1 b_0 \in \mathbb{B}^n$ represents the number $(u)_2 = \sum_{i=0}^{n-1} b_i 2^i$.) So P is meant to lead from register contents $\overline{u} = (u, \Box, \ldots)$ to contents $(u', \Box, \ldots)$ where $(u')_2 = (u)_2 + 1$.

Exercise 3 Let $\mathbb{A} = \{a_1, \ldots, a_r\}$ be an alphabet with r letters, $r \geqslant 1$, enumerated in this fixed order. The lexicographic ordering $<_{\text{lex}}$ on $\mathbb{A}^*$ is defined through:

$$w <_{\text{lex}} w' \text{ iff } \begin{cases} |w| < |w'| \text{ or} \\ |w| = |w'| \text{ and } w = ua_iv, w' = ua_jv' \text{ where } 1 \leqslant i < j \leqslant r. \end{cases}$$

A function $f \colon \mathbb{N} \to \mathbb{A}^*$ is *monotone* if $f(m) <_{\text{lex}} f(n)$ for all $m < n$. Show (informally, sketching corresponding algorithms in words if you like) that the following are equivalent for any $W \subseteq \mathbb{A}^*$ [you may want to refer to Lemma 4.1.2]:

(i) W is decidable.

(ii) W is finite or $W = \text{image}(f)$ for a monotone total recursive function $f \colon \mathbb{N} \to \mathbb{A}^*$.

Exercise 4 (cf. Lemma 4.1.2) Sketch a proof of the following claim for R-decidability and R-enumerability, i.e., for register machines over some fixed alphabet $\mathbb{A}$.
The following are equivalent for any $W \subseteq \mathbb{A}^*$:

(i) W is decidable.

(ii) Both W and $\overline{W} = \mathbb{A}^* \setminus W$ are recursively enumerable.

Exercise 5 Let $L_1, L_2 \subseteq \mathbb{A}^*$ be languages over the finite alphabet $\mathbb{A}$.

(a) Assume that $L_1 \subseteq L_2$. Discuss whether decidability/enumerability of L_1 or of L_2 have any implications on decidability/enumerability of L_2 or of L_1.

(b) What does decidability/enumerability of L_1 and L_2 imply about $L_1 \cup L_2$ and $L_1 \cap L_2$?

(c) Assume that $L_1 \subseteq L_2$ where both L_1 and L_2 are decidable and $L_2 \setminus L_1$ is infinite. Find an undecidable language L such that $L_1 \subseteq L \subseteq L_2$.
(Hint: you may use the undecidability of the halting problem.)

Exercise 6 [Extra] Sketch a procedure to simulate the operation of any register machine program using k registers, one of which is used as input/output register, by means of a register machine using only two registers. For simplicity, you may just consider $k = 3$.
Hint: working with an alphabet $\mathbb{A}' := \mathbb{A} \cup \{\#\}$ with a new letter $\#$, try to simulate the register content $\overline{u} = (u_1, \ldots, u_k)$ of P by $\overline{u}' := (u_1 \# u_2 \# \ldots \# u_k, \Box)$ of P'.