# Lab02-URM

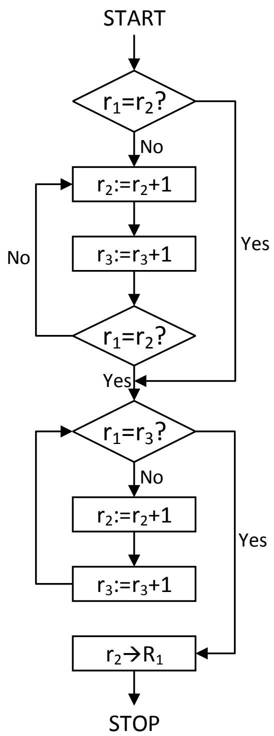
CS363-Computability Theory, Xiaofeng Gao

1. The right figure is the flow diagram for URM program $P$.

   (a) Write down the instructions of $P$.

   (b) Write down the progress of computation with initial configuration of $\{4, 2, 0, 0, \cdots\}$?

   (c) What is $f_P^{(2)}$ with initial configuration of $\{x, y, 0, 0, \cdots\}$




2. Devise URM programs to compute $f(x) = \max\{x, y\}$, and then draw the corresponding flow diagram.

3. Show "$x$ is even" is a decidable predicate on $\mathbb{Z}$.

4. Suppose $P$ is a program without any jump instructions. Show that

   (a) there is a number $m$ such that either $\forall x : f_P^{(1)}(x) = m$, or $\forall x : f_P^{(1)}(x) = x + m$.

   (b) not every computable function is computable in this sense.

5. Show that for each transfer instruction $T(m, n)$ there is a program without any transfer instructions that has exactly the same effect as $T(m, n)$ on any configuration of the URM (Thus transfer instructions are really redundant in the formulation of our URM; it is nevertheless natural and convenient to have transfer as a basic facility of the URM).

6. Gadgets

   In order to construct URM to perform complex operations, it is useful to build it from smaller components that we'll call *gadgets*, which perform specific operations. A gadget will be defined by a series of instructions and will operate on registers that are specified in the gadget's name. For instance, the gadget "predecessor $r_n$" denoted by $P(n)$ will subtract 1 from the contents of register $R_n$ if it is non-zero. It can be represented by an instruction sequence shown in the right block. For simplicity, when we obey gadget function $P(l)$, we by default obey $P^{-1}[l_1, \cdots, l_n \to l]$, meaning we will use registers $R_{l_1}, \cdots, R_{l_n}$ ($l_i > \rho(P)$, $\forall 1 \le i \le n$) and place the result in $R_l$, without any interference to the next instructions. Now answer the following questions:

Gadget $P(1)$

| | |
|---|---|
| $I_1$ | J(1,4,9) |
| $I_2$ | S(3) |
| $I_3$ | J(1,3,7) |
| $I_4$ | S(2) |
| $I_5$ | S(3) |
| $I_6$ | J(1,1,3) |
| $I_7$ | T(2,1) |

URM $Q$

| | |
|---|---|
| $I'_1$ | J(1,2,6) |
| $I'_2$ | S(2) |
| $I'_3$ | T(2,3) |
| $I'_4$ | M(2,3,4) |
| $I'_5$ | G(1,4,2) |
| $I'_6$ | H(2) |

   (a) Define a gadget "greater than $r_m > r_n$" denoted by $G(m, n, q)$, which determines whether the initial value of $R_m$ is greater than that of $R_n$. If yes, jump to the $q$th instruction, otherwise go on to the next instruction.

   (b) Define a gadget "halt with $r_n$" denoted by $H(n)$, which leaves $R_n$ with its initial value, and overwrites the initial values of other registers into 0 (write the instruction sequences).

   (c) Define a gadget "multiply $r_m$ by $r_n$ to $R_p$" denoted by $M(m, n, p)$, which multiplies $r_m$ by $r_n$ and stores the result in $R_p$.

   (d) Describe the function of one argument $f(x)$ computed by the program $Q$. (What is $f_Q^{(1)}$?)