recursive function is URM-computable

Canonical name	RecursiveFunctionIsURMcomputable
Date of creation	2013-03-22 19:04:51
Last modified on	2013-03-22 19:04:51
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	7
Author	CWoo (3771)
Entry type	Result
Classification	msc 03D10
Classification	msc 68Q05
Related topic	RecursiveFunction

Proposition 1. *Every recursive function is URM-computable function.*

The proof can be broken down in several simple steps.

Proposition 2. *The zero function, the successor function, and the projection functions are URM-computable.*

Proof. The zero function is computed by $Z(1)$, the successor function is computed by $S(1)$, and for any $n > 0$, the i-th projection function $p_i^n(x_1, \dots, x_n) = x_i$ is computed by $T(i, 1)$. $\square$

Proposition 3. *URM-computability is closed under functional composition.*

Proof. This is proved in the entry on combining URMs. $\square$

Proposition 4. *URM-computability is closed under primitive recursion.*

Proof. Suppose $f : \mathbb{N}^m \to \mathbb{N}, g : \mathbb{N}^{m+2} \to \mathbb{N}$ are computed by M, N respectively. Let $h : \mathbb{N}^{m+1} \to \mathbb{N}$ be obtained from f, g by primitive recursion, namely,

$$\begin{aligned} h(0, x_1, \dots, x_m) &:= f(x_1, \dots, x_m) \\ h(n+1, x_1, \dots, x_m) &:= g(h(x_1, \dots, x_m, n), n, x_1, \dots, x_m). \end{aligned}$$

Let P be the following URM:

$T(1, p+1), T(2, p+2), \cdots, T(m+1, p+m+1)$,
$M[p+2, \dots, p+m+1; p+m+2], J(p+1, p+m+3, q), S(p+m+3)$,
$N[p+2, \dots, p+m+1, p+m+3, p+m+2; p+m+2]$,
$J(1, 1, m+3), T(p+m+2, 1)$.

where $p = \max(m+2, \rho(M), \rho(N))$ and $q = m+7$. The program works as follows:

$\boldsymbol{I_1, \dots, I_{m+1}}$**:** transfer the first $m+1$ registers to another are on the tape:

$$T(1, p+1), T(2, p+2), \cdots, T(m+1, p+m+1)$$

$\boldsymbol{I_{m+2}}$**:** compute $h(0, x_1, \dots, x_m)$ using $M[p+2, \dots, p+m+1; p+m+2]$

$\boldsymbol{I_{m+3}}$: if the content of register $p+1$ (formerly the content of register 1) is the same as the content of $p+m+3$ (initially set to 0), go to the last instruction whose index is $q(=m+7)$; otherwise continue to the next instruction: $J(p+1, p+m+3, q)$

$\boldsymbol{I_{m+4}}$: increment register $p+m+3$ by 1: $S(p+m+3)$

$\boldsymbol{I_{m+5}}$: compute $h(i, x_1, \ldots, x_m)$, where i is the content of register $p+m+3$, using

$$N[p+2, \ldots, p+m+1, p+m+3, p+m+2; p+m+2]$$

$\boldsymbol{I_{m+6}}$: go to instruction $m+3$: $J(1, 1, m+3)$

$\boldsymbol{I_{m+7}}$: transfer result back to register 1: $T(p+m+2, 1)$.

Note that if $(x_1, \ldots, x_m, n) \in \operatorname{dom}(h)$, then $P(x_1, \ldots, x_m, n) \downarrow h(x_1, \ldots, x_m, n)$. Otherwise, $h(x_1, \ldots, x_m, n)$ is undefined. This can happen either $f(x_1, \ldots, x_m)$ is undefined, in which case M diverges, $g(x_1, \ldots, x_m, i, h(x_1, \ldots, x_m, i))$ is undefined, in which case N diverges, or $h(x_1, \ldots, x_m, i) \neq 0$ for all $i \in \mathbb{N}$, in which case P loops indefinitely. In all cases, P diverges. This shows that P computes h. $\square$

Proposition 5. *URM-computability is closed under minimization.*

Proof. Suppose $f : \mathbb{N}^{m+1} \to \mathbb{N}$ is computed by M. Let $g : \mathbb{N}^m \to \mathbb{N}$ be obtained from f by minimization. In other words, for any $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathbb{N}^m$, set

$$A(\boldsymbol{x}) := \{y \in \mathbb{N} \mid (z, \boldsymbol{x}) \in \operatorname{dom}(f) \text{ for all } z \leq y \text{ and } f(y, \boldsymbol{x}) = 0\}$$

and define

$$g(\boldsymbol{x}) := \begin{cases} \min A(\boldsymbol{x}) & \text{if } A(\boldsymbol{x}) \neq \varnothing, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Let Q be the following URM:

$$T(1, p+1), T(2, p+2), \cdots, T(m, p+m), M[p+m+1, p+1, \ldots, p+m; 1],$$
$$J(1, p+m+2, q), S(p+m+1), J(1, 1, m+1), T(p+m+1, 1)$$

where $p = \max(m+1, \rho(M))$ and $q = m+5$. The program works as follows:

$\boldsymbol{I_1, \ldots, I_m}$**:** transfer the first m registers to another are on the tape:

$$T(1, p+1), T(2, p+2), \cdots, T(m, p+m)$$

$\boldsymbol{I_{m+1}}$**:** compute $f(0, x_1, \ldots, x_m)$ using $M[p+m+1, p+1, \ldots, p+m; 1]$, where the content of register $p+m+1$ is set to 0 initially.

$\boldsymbol{I_{m+2}}$**:** if the content of register $p+m+2$ (which is always 0) is the same as the content of register 1 (value of $f(0, x_1, \ldots, x_m)$, if defined), go to the last instruction whose index is $q(= m+5)$; otherwise continue to the next instruction: $J(1, p+m+2, q)$

$\boldsymbol{I_{m+3}}$**:** increment register $p+m+1$ by 1 (counter): $S(p+m+1)$

$\boldsymbol{I_{m+4}}$**:** go to instruction $m+1$: $J(1, 1, m+1)$

$\boldsymbol{I_{m+5}}$**:** transfer content of register $p+m+1$ to register 1: $T(p+m+1, 1)$.

If $(x_1, \ldots, x_m) \in \mathrm{dom}(g)$, then $Q(x_1, \ldots, x_m) \downarrow g(x_1, \ldots, x_m)$. Otherwise, $g(x_1, \ldots, x_m)$ is undefined, which can happen either when $f(i, x_1, \ldots, x_m) \neq 0$ for all $i \in \mathbb{N}$, in which case Q loops indefinitely, or $f(i, x_1, \ldots, x_m)$ is undefined, while $f(j, x_1, \ldots, x_m)$ are defined and non-zero, for all $j < i$, in which case M diverges. In both cases, Q diverges. Hence Q computes g. $\square$

Since a recursive function is obtained by a finite application of functional operations specified in Propositions 3,4,5 on the basic arithmetic functions specified in Proposition 2, every recursive function is URM computable as result, proving Proposition 1.