More About Computability

Recall: We defined computable to mean "there is a URM program"

Wanted to execute two URM programs one after the other.

Cannot join "as is" because of jump addresses.

We now assume that all programs are in **standard form**:

A program $P = I_1 I_2 \ldots I_\ell$ is said to be in standard form if, for every instruction of the form $J(m,n,p)$ in P, $p \leq \ell+1$.

Exercise: Prove that every program can be cast in standard form.

In order to join even standard programs, we have to modify the addresses in the jump instructions in the second program.

A concatenation executes P followed by Q.

How does one ensure that the final configuration of P is an acceptable initial configuration for Q?

P is finite, so there is a smallest u s.t. P does not touch R_v for any $v > u$.
call this $\rho(P)$.

Make sure that Q only uses registers R_v, where $v > u$.

Clear that these two cases can be handled in general:

T1 If a program needs input to be given in registers $R_1, \ldots, R_n$ but input is provided in $R_{\ell_1}, \ldots, R_{\ell_n}$.

T2 If output needs to be written on R_ℓ instead of R_1.

after appropriate use of T1 and T2.

Concatenation: Let P and Q^* be programs of length ℓ and k. Then, the concatenation of P and Q (written PQ), is the program

$$I_1 I_2 \dots I_\ell \, I_{\ell+1} \dots I_{\ell+k}, \text{ where}$$

$P = I_1 I_2 \dots I_\ell$, and $I_{\ell+1} \dots I_{\ell+k}$ is the program obtained by ensuring that Q only modifies registers R_v where $v > \rho(P)$, and modifying every instruction of the form $J(m,n,p)$ in Q to $J(m,n,\ell+p)$

What about composing computable functions?

If f and g are computable, is $f \circ g$ computable?

Easy if f is unary (essentially, concat!) $f: \mathbb{N}^k \to \mathbb{N}$

Consider a k-ary f and n-ary $g_1, \ldots, g_k$, all computable. Then, h is computable, where $g_i: \mathbb{N}^n \to \mathbb{N}$

$$h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n))$$

Compute $g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)$,

then run the program for f on these values.

* Do not overwrite $x_1, \ldots, x_n$ during the computation of the g_is, and do not overwrite any g_i during the computation of g_j for $j > i$.

* Use $R_{m+1}, \dots, R_{m+n}$ to store $\bar{x}$, R_{m+n+i} to store $g_i(\bar{x})$, m "large".
Finally compute f on $r_{m+n+1}, \dots, r_{m+n+k}$, and store in R_1.

Composition: Let $F, G_1, \dots, G_k$ be programs computing $f, g_1, \dots, g_k$ resp.
Let $m = \text{sum}(n, k, \rho(F), \rho(G_1), \dots, \rho(G_k))$.
Store $x_1, \dots, x_n$ in $R_{m+1}, \dots, R_{m+n}$.
Run G_i on $x_1, \dots, x_n$, and store the result in R_{m+n+i}.
Then run F on $R_{m+n+1}, \dots, R_{m+n+k}$ as input.
This program computes h.

Similarly, the following operations preserve computability.

Rearrangement: Suppose f is a k-ary computable function, and $x_{i_1}, \dots, x_{i_k}$ is some sequence formed using variables from $x_1, \dots, x_n$. Then, if g is the function given by

$$g(x_1, \dots, x_n) = f(x_{i_1}, \dots, x_{i_k}),$$

g is computable. Projection + Composition

Similarly for other definitions of g as follows:

Identification: $g(x) = f(x, \dots, x)$

Adding unused variables: $g(x_1, \dots, x_k, x_{k+1}) = f(x_1, \dots, x_k)$.

* Is f computable, where $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$?

$$f(x_1, x_2, x_3) = h(h(x_1, x_2), x_3), \text{ where}$$

Computable, by Composition

$$h(x, y) = x + y$$

Computable, shown earlier

* Is f computable, where $f(x) = s$, for some $s \in \mathbb{N}$?

* Is f computable, where $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$?

$$f(x_1, x_2, x_3) = h(\underbrace{h(x_1, x_2)}_{\text{Computable, by Composition}}, x_3), \text{ where}$$

$$\underbrace{h(x, y) = x + y}_{\text{Computable, shown earlier}}$$

* Is f_s computable, where $f_s(x) = s$, for some $s \in \mathbb{N}$?

* Suppose $f(-,-)$ is computable, and $s \in \mathbb{N}$. Is g computable, where $g(x) = f(x, s)$?

What about the following function?

$$f(0) = 0$$

$$f(n+1) = f(n)+1.$$

Is f computable?

Recursion: Let $x = (x_1, \ldots, x_n)$, and let f and g be n-ary and $(n+2)$-ary computable functions. Then, there is a unique $(n+1)$-ary function h as follows

$$h(x_1, \ldots, x_n, 0) = f(x_1, \ldots, x_n)$$

$$h(x_1, \ldots, x_n, y+1) = g(x_1, \ldots, x_n, y, h(x_1, \ldots, x_n, y))$$

and h is computable.

Example: $f() = 1$, and $g(y, z) = z(y+1)$

What does h compute?

$$h(0) = f() = 1$$

$$h(y+1) = g(y, h(y)) = h(y) \cdot (y+1)$$