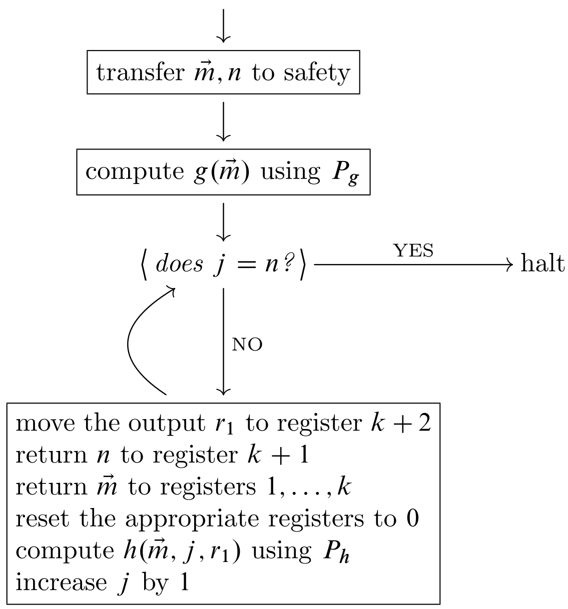
Primitive recursion is URM-computable

Let P_g compute $g(\vec{m})$ and P_h compute $h(\vec{m}, n, p)$, where $\vec{m}$ is of fixed length $k \geq 0$. We want a program P_f that computes the function f defined by

$$f(\vec{m}, 0) = g(\vec{m}),$$
$$f(\vec{m}, n+1) = h(\vec{m}, n, f(\vec{m}, n)).$$

Let ρ be the largest register referred to by P_g or P_h. We need to preserve the input $(\vec{m}, n)$ of f, so we will store those in registers $\rho+1, \ldots, \rho+k+1$. We also need to keep track of how many times we have done the recursive step defined by h; let's call this number j and store it in register $\rho+k+2$. Note that initially $j = 0$, which is what we want.

Here's the idea:



We can achieve this with the following program (which may not be optimal):

(1)	$T(1, \rho+1)$
$\vdots$	$\vdots$
$(k+1)$	$T(k+1, \rho+k+1)$
$(k+2)$	$Z(k+1)$
$(k+3) \cdots (k+3+\ell(P_g))$	P_g
$(k+4+\ell(P_g))$	$J(\rho+k+1, \rho+k+2, k+\rho+8+\ell(P_g)+\ell(P_h))$
$(k+5+\ell(P_g))$	$T(1, k+2)$
$(k+6+\ell(P_g))$	$T(\rho+k+1, k+1)$
$\vdots$	$\vdots$
$(2k+6+\ell(P_g)$	$T(\rho+1, 1)$
$(2k+7+\ell(P_g)) \cdots (k+\rho+5+\ell(P_g))$	$Z[k+3, \rho]$
$(k+\rho+5+\ell(P_g)+1) \cdots (k+5+\rho+\ell(P_g)+\ell(P_h))$	P_h
$(k+\rho+6+\ell(P_g)+\ell(P_h))$	$S(\rho+k+2)$
$(k+\rho+7+\ell(P_g)+\ell(P_h))$	$J(1, 1, k+4+\ell(P_g))$

Thus, we use $k+\rho+7$ additional steps beyond the lengths of P_g and P_h. This shows that the URM-computable functions are closed under the primitive recursion scheme.