Práctica 2:

① • $V_i \leftarrow k$ // supongo k está en X_1.

Código:

```
    Y ← 0
[A] IF X1 ≠ 0 GOTO L
    GOTO E          // VAMOS A EXI
[L] Y ← Y + 1
    X1 ← X1 - 1
    IF X1 ≠ 0 GOTO A
```

• GOTO L

Código:

```
Z1 ← Z1 + 1     // Z1 variable fresca, inicia en 0
IF Z1 ≠ 0 GOTO L
```

• IF $V_i = 0$ GOTO L

Código:

```
IF Vi ≠ 0 GOTO E
GOTO L
```

(B) $f(x_1, x_2) = x_1 + x_2$

```
     Y ← X1
     Z1 ← X2
[L]  IF Z1 = 0 GOTO E
     Y ← Y + 1
     Z1 ← Z1 − 1
     GOTO L
```

(C)

$\psi_P^{(1)}(x) = x. \longrightarrow$ computa $f(x, 0)$

$\psi_P^{(2)}(x_1, x_2) = x_1 + x_2 \longrightarrow$ computa $f(x_1, x_2)$

$\psi_P^{(3)}(x_1, x_2, x_3) = x_1 + x_2 \longrightarrow$ computa $f(x_1, x_2)$

La tercera entrada es ignorada xq en el Programa P jamás se usa X_3. Es una función de 2 variables.

(2)

(A) Tenemos que ver que C_S está cerrado por composición y recursión Primitiva.

→ Veamos primero las funciones iniciales:

- $N(x) = 0 \iff Y \leftarrow 0$, "PROGRAMA VACÍO".
- $s(x) = x + 1 \iff Y \leftarrow X_1 + 1$.
- $u_i^N(x_1, \dots, x_N) = x_i \iff Y \leftarrow X_i$

→ Veamos ahora que está cerrada por composición:

Sean $f, g_1, \dots, g_k \in C_S$. El sig. Programa P:

k VECES

$$Z_1 \leftarrow g_1(X_1, \dots, X_N)$$
$$Z_2 \leftarrow g_2(X_1, \dots, X_N)$$
$$\vdots$$
$$Z_k \leftarrow g_k(X_1, \dots, X_N)$$
$$Y \leftarrow f(Z_1, \dots, Z_k)$$

computa a la función h: $h(X_1, \dots, X_N) = f(g_1(X_1, \dots, X_N), \dots, g_k(X_1, \dots, X_N))$

que pertenece a C_S. $(h \in C_S)$.

Entonces C_S está cerrada x composición.

→ Veamos que está cerrado por recursión primitiva.

Sean $f, g \in C_S$. El sig. Prog. P:

$Z_1 \leftarrow X_{N+1}$

$Y \leftarrow f(x_1, \dots, x_N)$

$[L]$ IF $Z_1 = 0$ GOTO E

$Z_2 \leftarrow Z_2 + 1$

$Z_1 \leftarrow Z_1 - 1$

$Y \leftarrow g(Y, X_1, \dots, X_N, Z_2)$

GOTO L

computa la función $h \in C_S$ que se obtiene por r.p.

→ C_S es una clase PRC.

B) Probamos que la multiplicación es p.r. en la práctica 1. Sabemos que toda función p.r. es computable.

C) Es TOTAL-COMPUTABLE.

3) A) $\Psi_P^{(N)} = \Psi_{P'}^{(N)}$ $\forall N \geq 1$ $\iff$ 2 programas distintos que tienen el mismo output para cualquier entrada.

→ No sé demostrarlo de forma formal.

Si P no es autocontenido, existe un programa P' que incluye a la etiqueta no contenida en P, y en esa línea tiene una instrucción que no afecta el output.

B) • IF $r(V_1, \dots, V_N)$ GOTO L

$Z_1 \leftarrow r(V_1, \dots, V_N)$
IF $Z_1 \neq 0$ GOTO L

- IF $r(V_1, \dots, V_N)$ THEN P ELSE Q

$Z_1 \leftarrow r(V_1, \dots, V_N)$
IF $Z_1 \neq 0$ GOTO P
GOTO Q

(c) Es fácil ver que si $x \geq 5$ el programa se indefine.

$$\left.\begin{array}{ll} & \text{IF } X_1 = 3 \text{ GOTO } F \\ [L] & \text{IF } X_1 \geq 5 \text{ GOTO } L \\ & Y \leftarrow 2 \cdot X_1 \\ & \text{GOTO } E \\ [F] & Y \leftarrow 1 \end{array}\right] \; h(x).$$

(4) (A)

- NO HAY MULTIPLICACIÓN.
- NO SE PUEDE HACER $f(x, y) = x + y$.
- NO HAY DIVISIÓN.

No pienso hacer una demostración formal.

(B) $\Longrightarrow$) Fácil de ver. Si la función parcial computable está en S', también está en S ya que tiene las mismas instrucciones (y la resta).

$\Longleftarrow$) CONSULTAR

5 A Estamos usando minimización no acotada. Puede suceder que no haya un t que cumpla, por lo que la función nunca termina.

$$Z_1 \leftarrow X_{n+1}$$

$[L]$ IF $P(X_1, \dots, X_n, Z_1)$ GOTO E

$Z_1 \leftarrow Z_1 + 1$

GOTO L

$[E]$ $Y \leftarrow Z_1$

NOS QUEDAMOS ATRAPADOS SI NINGÚN Z_1 CUMPLE.

B CONSULTAR.

6 Tengo que chequear, en el caso de un IF, si su etiqueta aparece en una instrucción anterior o no.

$$r(x) = (\forall i)_{\leq |x+1|} \left(\neg (\exists j)_{\leq i} \; \ell(r(x+1[i])) \dot{-} 2 = \ell(x+1[j]) \wedge \ell(x+1[j]) > 0 \right)$$

$\rightarrow$ $\boxed{\ell(r(x+1[i]))} \rightarrow \#(I)_i \rightarrow \langle a, \langle b, c \rangle\rangle = \boxed{b}$

$\rightarrow$ $\boxed{\ell(x+1[j])} \rightarrow \#(I)_j \rightarrow \langle a, \langle b, c \rangle\rangle = \boxed{a}$

"DESMENUZANDO LA FUNCIÓN".

Si $a = b - 2$, coincide la etiqueta.

Pido que a (y b) no sean 0 xq eso significa que en esa posición (INSTRUCCIÓN) no hay una etiqueta.

$\rightarrow$ Quiero evitar la situación $b = 0$ $(V \leftarrow V)$ y $a = 0$ (INSTRUCCIÓN j SIN ETIQUETA).

(7) • $f_1 \rightarrow y \in Dom\, \Phi_x = \Phi_x(y)\downarrow$

$f_1(x,y) = (\exists t)\ STP(y, x, t)$

↳ step es P.r., entonces podemos usar el existencial no acotado.

• f_2

$f_2(x) = (\exists \langle t, z \rangle)\ STP(z, x, t)$

step retorna 1 si el programa termina en un tiempo t para una entrada z. $\rightarrow z \in Dom\, \Phi_x$

• $f_3 \rightarrow y \in Im\, \Phi_x = \Phi_x(z) = y$ PARA ALGÚN z.

$f_3(x,y) = (\exists \langle t, z \rangle)\left(STP(z,x,t) \wedge r(SNAP(z,x,t))[1] = y\right)$

SNAP = < #(1), LISTA DEL ESTADO DEL PROGRAMA >

↳ Por orden, el primero SIEMPRE es Y.

• $f_4 \rightarrow$ coincide al menos la entrada válida de un programa x con la salida de un programa y.

$f_4(x,y) = (\exists t)\ (f_1(x,t) \wedge f_3(y,t))$

f_1 y f_3 son parcial computables. $\therefore$ f_4 ES PARCIAL COMPUTABLE.

(8)

(A) IDEA: Poder acotar por $Q(\log x)$ un existencial.

Si componemos funciones p.r., f es p.r.

Por hipótesis, sabemos que f es parcial computable en tiempo polinomial $\longrightarrow$ Existe Q que acota a f.

$$\boxed{f(x) = (\exists t)_{\leq Q(\log_2 x)}\ STP(x, \#P, t)}$$

(B) Sí. Si la cota es P.r., entonces "hace" P.r. a f.

(C)

(9) "... la variable de salida Y luego de la ejecución de P con entradas $x_1, \dots, x_N$ vale $\#P$".

$\longrightarrow$ No implica que P haya terminado.

$$\boxed{f_m(x) = (\exists t) \quad r(\text{SNAP}(\underbrace{x_1, \dots, x_N}, x, t))[1] = x}$$

ESTO ES LO QUE YO ENTIENDO POR N ENTRADAS $x_1, \dots, x_n$.

$\longrightarrow$ PREGUNTAR x LAS DUDAS.