Codificación de programas, STP y SNAP

Ariel Bendersky

Febrero 2018

Codificación de programas - Mini repaso

Las instrucciones de $\mathcal{S}$ son:

1. $V \leftarrow V + 1$
2. $V \leftarrow V - 1$
3. $IF\ V \neq 0\ GOTO\ L'$
4. $V \leftarrow V$ (La agregamos por conveniencia. Es evidente que no agrega poder de cómputo).

Observaciones:

- Toda instrucción puede o no tener una etiqueta L.
- Toda instrucción menciona exactamente una variable.
- El *GOTO* menciona, además, una etiqueta para el salto.

Codificación de programas - Mini repaso

Para codificar una instrucción I la metemos en la terna $\#I = < a, < b, c >>$, donde:

- Si I tiene etiqueta L, entonces $a = \#L$ (alfabéticamente). Si no, $a = 0$-
- Si la variable mencionada es V entonces $c = \#V - 1$ (ordenadas $Y, X_1, Z_1, X_2, Z_2, ...$).
- Si la instrucción es $V \leftarrow V$ entonces $b = 0$.
- Si la instrucción es $V \leftarrow V + 1$ entonces $b = 1$.
- Si la instrucción es $V \leftarrow V - 1$ entonces $b = 2$.
- Si la instrucción es $IF\ V \neq 0\ GOTO\ L'$ entonces $b = \#L' + 2$.

Un programa es una lista de instrucciones $\#P = [\#I_1, \#I_2, ...] - 1$.

Step counter - Mini repaso

$$STP^{(n)}(x_1, ..., x_n, \#P, t)$$

es un predicado PR (para cada $n > 0$) que nos dice si el programa P termina en tiempo menor o igual a t con entradas $x_1, ..., x_n$

Snapshot - Mini repaso

$$< i, \sigma >= SNAP^{(n)}(x_1, ..., x_n, \#P, t)$$

es una función PR (para cada $n > 0$) que nos devuelve la descripción instantánea del programa después de t pasos. i nos dice el número de línea que hay que ejecutar y σ es una lista con los valores de todas las variables que aparecen en P después de t pasos $[Y, X_1, Z_1, X_2, Z_2, ...]$.

Problema 1

Decimos que un programa es *mala onda* si no tiene instrucciones de la forma:

$$X_i \to X_i + 1$$

Es decir, no aumenta el valor de las entradas.
Demuestre que es PR el predicado:

$$p(x) = \begin{cases} 1 & \text{Si el programa de número } x \text{ es mala onda} \\ 0 & \text{Otro caso} \end{cases}$$

Resolución

Uso un para todo acotado. Quiero que para toda línea, si es del tipo "incrementar 1", la variable a la que hace referencia sea una variable en posición impar (recordemos que se ordenan como $Y, X_1, Z_1, X_2, Z_2, ...$)

$$p(x) = \forall_{t \leq |x+1|} (l(r((x+1)[t])) = 1 \rightarrow (1 - Par(r(r((x+1)[t])))))$$

Siempre miro $x + 1$ es porque la codificación de programas le resta 1 a la lista.

Problema 2

Decimos que un programa P *se cuelga a lo bobo* si posee dos instrucciones consecutivas de la forma:

$$V \leftarrow V + 1$$

$$[L] IF\ V \neq 0\ GOTO\ L$$

Demuestre que es PR el predicado:

$$p(x) = \begin{cases} 1 & \text{Si el programa de número } x \text{ se cuelga a lo bobo} \\ 0 & \text{Otro caso} \end{cases}$$

Resolución

Es un existencial acotado sobre t (menor al largo del programa menos uno) que cumpla lo siguiente:

- La instrucción t es de la forma $V \leftarrow V + 1$. Esto es: $l(r((x+1)[t])) = 1$.
- La instrucción $t+1$ es de la forma $[L] IF\ W \neq 0\ GOTO\ L'$. Es decir: $l(r((x+1)[t+1])) > 2$.
- $L' = L$. Es decir: $l(l((x+1)[t+1])) = l(r(x+1)[t+1]) - 2$
- $V = W$. Se traduce en: $r(r((x+1)[t])) = r(r((x+1)[t+1]))$.

Resolución

Juntamos todo:

$$p(x) = \exists_{t\leq|x+1|-1}(l(r((x+1)[t])) = 1)\wedge(l(r((x+1)[t+1])) > 2)\wedge$$

$$\wedge(l(l((x+1)[t+1])) = l(r(x+1)[t+1]) - 2)\wedge$$

$$\wedge(r(r((x+1)[t])) = r(r((x+1)[t+1])))$$

Otro tema

Minimización acotada vs. no acotada

Acotada:

$$\min_{t\leq y} p(t, x_1, \ldots, x_n) = \begin{cases} \text{Mínimo } t \leq y \text{ tal que} & \\ p(t, x_1, \ldots, x_n) & \text{si existe} \\ & \\ 0 & \text{si no.} \end{cases}$$

Es PR si $p(t, x_1, \ldots, x_n)$ es PR.

Minimización acotada vs. no acotada

No acotada:

$$\min_t p(t, x_1, \dots, x_n) = \begin{cases} \text{Mínimo } t \text{ tal que} & \\ p(t, x_1, \dots, x_n) & \text{si existe} \\ & \\ \uparrow & \text{si no.} \end{cases}$$

Es parcial computable si $p(t, x_1, \dots, x_n)$ es computable.

Problema 3

Demuestre que es parcial computable la siguiente función:

$$f(x, y) = \begin{cases} 1 & \text{si } \Phi_x^{(1)}(y)\downarrow \text{ y } \Phi_x^{(1)}(y) < x \\ \uparrow & \text{si no.} \end{cases}$$

Solución en $\mathcal{S}$

Uso la macro del interprete universal.

$$Y \leftarrow Y + 1$$

$$Z_1 \leftarrow \Phi^{(1)}_{X_1}(X_2)$$

$$Z_2 \leftarrow Z_1 \geq X_1$$

$$[L] IF\ Z_2 \neq 0\ GOTO\ L$$

Solución con minimización no acotada

$$g(x, y) = \min_t \left(STP^{(2)}(y, x, t) \wedge r(SNAP^{(n)}(x, y, t))[1] < x \right)$$

Como el argumento es PR, es también computable. Luego, la minimización es parcial computable. Finalmente, considero la función computable $h(x) = 1$ y computo $f(x, y) = h(g(x, y))$.

Problema 4 (para lxs que siguen despiertxs)

$$f(x,y) = \begin{cases} 1 & \text{si existe z tal que} \\ & \Phi_x^{(1)}(z)\downarrow, \Phi_y^{(1)}(z)\downarrow, \Phi_x^{(1)}(z) = \Phi_y^{(1)}(z) \\ \uparrow & \text{si no} \end{cases}$$

Solución con minimización no acotada

$$min_{<t,z>}(STP^{(1)}(z,x,t) \wedge STP^{(1)}(z,y,t)\wedge$$
$$\wedge r(SNAP^{(1)}(z,x,t))[1] = r(SNAP^{(1)}(z,y,t))[1])$$

y de nuevo lo paso por la función constante 1.