Cartilla del lenguaje $\mathcal{S}$

Paradigmas de Lenguajes de Programación, 1^{er} cuat. 2021

1. Variables

Los valores que pueden tomar las variables son números enteros no negativos.

- Variables de entrada (infinitas): $X_1, X_2, X_3, \ldots$
- Variable de salida (única): Y
- Variables temporales (infinitas): $Z_1, Z_2, Z_3, \ldots$

2. Instrucciones

Hay cuatro tipos de instrucciones:

- $V \leftarrow V + 1$: representa el incremento del valor de la variable V en una unidad.
- $V \leftarrow V - 1$: si el valor de V es 0, no produce cambios. Si no, decrementa V en una unidad.
- $\mathsf{IF}\, V \neq 0\, \mathsf{GOTO}\, L$: si el valor de V es distinto a 0, ejecutar a continuación la instrucción etiquetada con L. En caso contrario continuar la ejecución en la instrucción de la línea siguiente.
- $V \leftarrow V$: es una instrucción sin efecto, se agrega por conveniencia.

Las instrucciones pueden o no estar etiquetadas. Por ejemplo, la siguiente instrucción está etiquetada con B:

$$[B]\, Z \leftarrow Z - 1$$

3. Programas

Un programa escrito en $\mathcal{S}$ consiste de una secuencia finita de instrucciones. La longitud de un programa es igual a la cantidad de instrucciones, siendo el programa vacío de longitud 0.

4. Ejecución

El valor inicial para las variables temporales y la variable de salida es 0. Las instrucciones se ejecutan consecutivamente (excepto cuando hay saltos).

5. Estado

El estado de un programa es una lista de ecuaciones de la forma $V = m$, donde V es una variable y m es un número entero no negativo. La lista contiene exactamente una ecuación para cada variable que aparece en el programa. El estado representa el valor de cada variable en un momento dado de la ejecución del programa.

6. Descripción instantánea

Dado un estado y un programa, para saber qué es lo que pasa "a continuación" en la ejecución, es necesario saber qué instrucción se va a ejecutar en el próximo paso.

- Si P es un programa de longitud n, llamamos descripción instantánea (o snapshot) de P a la tupla (i, e), donde $1 \leqslant i \leqslant n + 1$ y e es un estado de P. i simboliza al número de la instrucción a ser ejecutada a continuación, donde $i = n + 1$ significa que la ejecución terminó.

- Dada V una variable de P y una descripción instantánea (i, e), llamamos "valor de V en (i, e)" al número asociado a la variable V en el estado e.

- Una descripción instantánea (i, e) de P es terminal cuando $i = n + 1$, donde n es la longitud de P. Si (i, e) no es terminal, definimos a su sucesor (i', e') como:

  Caso 1 : si la i-ésima instrucción de P es $V \leftarrow V + 1$ y e contiene la ecuación $V = m$. Entonces $i' = i + 1$ y e' se obtiene reemplazando en e la ecuación $V = m$ por $V = m + 1$.

  Caso 2 : si la i-ésima instrucción de P es $V \leftarrow V - 1$ y e contiene la ecuación $V = m$. Entonces $i' = i + 1$ y e' se obtiene reemplazando en e la ecuación $V = m$ por $V = m - 1$ si $m \neq 0$, y si $m = 0$ se tiene $e' = e$.

  Caso 3 : si la i-ésima instrucción de P es $V \leftarrow V$. Entonces $i' = i + 1$ y $e' = e$.

  Caso 4 : si la i-ésima instrucción de P es $\mathsf{IF}\ V \neq 0\ \mathsf{GOTO}\ L$. Entonces $e' = e$, y hay tres subcasos:

  - Si e contiene la ecuación $V = 0$, entonces $i' = i + 1$.
  - Si e contiene la ecuación $V = m$, con $m \neq 0$, y existe en P al menos una instrucción con la etiqueta L, donde j es el mínimo número tal que la j-ésima instrucción está etiquetada con L, entonces $i' = j$. Es decir, si existen dos instrucciones etiquetadas con L en P, tomamos la primera.
  - En otro caso, $i' = n + 1$.

7. Cómputo

Un cómputo de un programa P se define como una secuencia de descripciones instantáneas de P

$$s_1, s_2, s_3, \ldots, s_k$$

tales que s_{i+1} es sucesor de s_i para $i \in \{1, 2, \ldots, k - 1\}$ y s_k es terminal.

8. Descripción instantánea inicial

Sea P un programa, y sean $r_1, r_2, \ldots, r_m$ m números dados (de entrada). Se define el estado inicial e de P para $r_1, r_2, \ldots, r_m$ como el que contiene las ecuaciones:

$$[X_1 = r_1, X_2 = r_2, \ldots, X_m = r_m, Y = 0]$$

Junto con las ecuaciones $V = 0$ para cada variable V que aparezca en P y sea distinta a $X_1, X_2, \ldots, X_m, Y$.
La descripción inicial de P para $r_1, r_2, \ldots, r_m$ es entonces la tupla $(1, e)$.

9. Cómputo a partir del estado inicial

Sea P un programa, y sean $r_1, r_2, \ldots, r_m$ números dados. Se define al cómputo a partir del estado inicial de P para $r_1, r_2, \ldots, r_m$ al cómputo de P

$$s_1, s_2, s_3, \ldots, s_k$$

donde s_1 es la descripción instantánea inicial de P para $r_1, r_2, \ldots, r_m$.

10. Codificación

10.1. Codificación de pares

Se define la siguiente función:

$$\langle x, y \rangle = 2^x(2y+1) - 1$$

Dado z cualquier número, existe una solución única para la ecuación $\langle x, y \rangle = z$, donde x es el mayor número tal que $2^x \mid z+1$ e y es la solución de la ecuación $2y+1 = \frac{z+1}{2^x}$.
Se definen los observadores del par como $x = l(z)$, $y = r(z)$, y se tiene el siguiente resultado:

$$l(z) = \min_{x \leqslant z}\{(\exists y)_{\leqslant z} : z = \langle x, y \rangle\}$$

$$r(z) = \min_{y \leqslant z}\{(\exists x)_{\leqslant z} : z = \langle x, y \rangle\}$$

10.2. Codificación de secuencias

Se define al número de Gödel de la secuencia $(a_1, \ldots, a_n)$ como:

$$[a_1, \ldots, a_n] = \prod_{i=1}^{n} p_i^{a_i}$$

donde p_i es el i-ésimo primo ($p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$, ...). Notar que con esta representación no podemos representar listas con ceros al final.

10.3. Codificación de programas

Vamos a asociar un número a cada programa: si P es un programa del lenguaje $\mathcal{S}$, llamamos $\#P$ al número que lo codifica.
Primero se define el siguiente orden para las variables:

$$Y\ X_1\ Z_1\ X_2\ Z_2\ X_3\ Z_3 \ldots$$

Se define el siguiente orden para las etiquetas:

$$A\ B\ C\ D \ldots AA\ AB\ AC \ldots AZ\ BA\ BB \ldots$$

Definimos como $\#V$ y $\#L$ la posición de la variable o etiqueta de acuerdo a su orden respectivo, comenzando desde 1. Por ejemplo, en este caso $\#Z_1 = 3$ y $\#B = 2$.
Sea I una instrucción (etiquetada o no) del lenguaje $\mathcal{S}$. Entonces definimos $\#I = \langle a, \langle b, c\rangle\rangle$, donde:

- si I no está etiquetada, entonces $a = 0$, y si está etiquetada con L entonces $a = \#L$,
- si la (única) variable que aparece en I es V, entonces $c = \#V - 1$,
- si la instrucción corresponde a $V \leftarrow V$, $V \leftarrow V + 1$ o $V \leftarrow V - 1$, se tiene $b = 0$, 1 o 2 respectivamente,
- si la instrucción corresponde a $\mathsf{IF}\, V \neq 0\, \mathsf{GOTO}\, L'$ entonces $b = \#(L') + 2$.

Ejemplos de codificación de instrucciones

- El número correspondiente a la instrucción sin etiquetar $X_1 \leftarrow X_1 + 1$ es

$$\langle 0, \langle 1, 1\rangle\rangle = \langle 0, 5\rangle = 10$$

- El número correspondiente a la instrucción etiquetada [A] $X_1 \leftarrow X_1 + 1$ es

$$\langle 1, \langle 1, 1\rangle\rangle = \langle 1, 5\rangle = 21$$

Notar que la codificación de instrucciones es una función biyectiva de instrucciones a números naturales.
Finalmente, sea P un programa que consiste de las instrucciones $I_1, I_2, \ldots, I_k$. Entonces definimos

$$\#P = [\#(I_1), \#(I_2), \ldots, \#(I_k)] - 1$$

Ejemplo de codificación de programas

Sea P el siguiente programa:

$$[A]\, X_1 \leftarrow X_1 + 1 \qquad (I_1)$$

$$\mathsf{IF}\, X_1 \neq 0\, \mathsf{GOTO}\, A \qquad (I_2)$$

Ya vimos antes que $\#(I_1) = 21$. Por otra parte $\#(I_2) = \langle 0, \langle 3, 1\rangle\rangle = \langle 0, 23\rangle = 46$. Finalmente se tiene que

$$\#P = [21, 46] - 1 = 2^{21} * 3^{46} - 1$$

Notar que la codificación de la instrucción sin etiquetar $Y \leftarrow Y$ es $\langle 0, \langle 0, 0 \rangle \rangle = \langle 0, 0 \rangle = 0$. Entonces, por la ambigüedad de los números de Gödel, el número de un programa no cambia si se le agrega la instrucción $Y \leftarrow Y$ al final. Para resolver esto introducimos la restricción de que la última instrucción de un programa no puede ser la instrucción sin etiquetar $Y \leftarrow Y$.
De esta manera cada número codifica un único programa y el programa vacío se codifica con el número 0.