Capítulo IV:
FUNCIONES RECURSIVAS

IV.5: PROGRAMAS UNIVERSALES

Mario de J. Pérez Jiménez
Grupo de investigación en Computación Natural
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Lógica Matemática

Curso 2010-11

Codificación de programas GOTO

Se va a codificar cada programa GOTO por un número natural:

- Codificaremos las variables.
- Codificaremos las etiquetas.
- Codificaremos las instrucciones GOTO (usando la función par).
  - Todo número natural codifica una única instrucción GOTO.
- Codificaremos los programas GOTO (usando la función de Gödel).
  - Todo número natural codifica un único programa GOTO.

Codificación de las Instrucciones (I)

Una instrucción consta básicamente de tres elementos:

- Etiqueta.
- Variable.
- Formato (o tipo) de la instrucción.

Codificación de las etiquetas:

Ordenamos las etiquetas: $A_1, B_1, C_1, D_1, E_1, A_2, ..., E_2, A_3, ...E_3, \dots$
Asignamos un número a cada etiqueta según este orden:

$$(k \geq 1) : \begin{cases} \#(A_k) = 5(k-1)+1 \\ \#(B_k) = 5(k-1)+2 \\ \#(C_k) = 5(k-1)+3 \\ \#(D_k) = 5(k-1)+4 \\ \#(E_k) = 5k \end{cases}$$

Codificación de las Instrucciones (II)

Codificación de las variables:

Ordenamos las variables: $Y, X_1, Z_1, X_2, Z_2, \cdots$
Asignamos un número a cada variable según este orden:

$$\begin{cases} \#(Y) & = 1 & \\ \#(X_k) & = 2k & (k \geq 1) \\ \#(Z_k) & = 2k+1 & (k \geq 1) \end{cases}$$

Codificación del Formato:

$$\begin{cases} 0 & \textit{si el formato es } V \longleftarrow V \\ 1 & \textit{si el formato es } V \longleftarrow V+1 \\ 2 & \textit{si el formato es } V \longleftarrow V-1 \\ \#(L)+2 & \textit{si el formato es IF } V \neq 0 \textit{ GOTO } L \end{cases}$$

Codificación de las Instrucciones (III)

Si I es una instrucción `GOTO`, entonces se define:

$$\#(I) = \langle a, \langle b, c \rangle \rangle$$

en donde:

$$a = \begin{cases} 0 & \textit{si } I \textit{ no tiene etiqueta} \\ \#(L) & \textit{si } I \textit{ tiene etiqueta } L \end{cases}$$

$$b = \textit{Código del formato de } I$$

$$\begin{cases} 0 & \textit{si formato } V \longleftarrow V \\ 1 & \textit{si formato } V \longleftarrow V + 1 \\ 2 & \textit{si formato } V \longleftarrow V - 1 \\ \#(L) + 2 & \textit{si formato } IF\ V \neq 0\ GOTO\ L \end{cases}$$

$$c = \#(V) - 1 \qquad \textit{si } V \textit{ es la variable de } I.$$

- Todo número natural $n \in \mathbb{N}$ codifica una única instrucción `GOTO`.

Codificación de los Programas

Si $P = (I_1, I_2, \ldots, I_n)$ es un programa GOTO, entonces se define:

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

Nota: Si $P_\emptyset$ es el programa vacío, entonces $\#(P_\emptyset) = [\varepsilon] - 1 = 0$.

Observaciones:

- Puesto que $\#(Y \longleftarrow Y) = \langle 0, \langle 0, 0 \rangle \rangle = 0$ y $[a_1, ..., a_n] = [a_1, ..., a_n, 0]$, se tiene:

  $$[\#(I_1), ..., \#(I_n)] = [\#(I_1), ..., \#(I_n), \#(Y \longleftarrow Y)]$$

  Por tanto, si los programas GOTO pudiesen terminar con la instrucción $Y \longleftarrow Y$ la codificación de programas no sería inyectiva.
- Todo número natural $n \in \mathbb{N}$ codifica un único programa GOTO.

Programas universales (I)

Para cada $n \geq 1$,

- ¿es computable la función de aridad $(n+1)$ que asigna a cada n-tupla $(x_1, \ldots, x_n)$ y cada programa GOTO P (codificado por $\#(P)$) el resultado de la computación $P(x_1, \ldots, x_n)$?

Responderemos afirmativamente a esta cuestión.

- Un programa U_n que calcule esa función de aridad $(n+1)$ se denomina *programa universal*.
- La función de aridad $(n+1)$ que calcula U_n se notará por $\mathcal{U}_n$.
- Así pues: $[\![U_n]\!]^{(n+1)}(\vec{x}, \#(P)) = [\![P]\!]^{(n)}(\vec{x}) = \mathcal{U}_n(\vec{x}, \#(P))$

Teorema. Para cada $n \geq 1$, existe un programa GOTO U_n tal que para todo $\vec{x} \in \mathbb{N}^n$ y todo programa GOTO, P, se tiene:

$$[\![U_n]\!]^{(n+1)}(\vec{x}, \#(P)) = [\![P]\!]^{(n)}(\vec{x})$$

Básicamente, un programa universal es un *intérprete de* GOTO.

Programas universales (II)

Codificación de configuraciones.

- *Estados*: Si $\sigma = \{V_1 = r_1, V_2 = r_2, ..., V_n = r_n\}$ es un estado, con $\#(V_i) = i$, se define:
$$\#(\sigma) = [r_1, ..., r_n]$$

- *Configuraciones*: Si (i, σ) es una configuración, se define:
$$\#(i, \sigma) = \langle i, \#(\sigma) \rangle$$

El programa universal U_n

$$\left.\begin{array}{l} Z \longleftarrow X_{n+1}+1 \\ S \longleftarrow \prod_{i=1}^{n}(p_{2i})^{X_i} \\ K \longleftarrow 1 \end{array}\right\} (1)$$

inicialización

$$[C] \quad IF\ K = Long(Z)+1\ GOTO\ F\ \} (2)$$

test de parada

$$\left.\begin{array}{l} U \longleftarrow r((Z)_k) \\ P \longleftarrow p_{r(U)+1} \\ IF\ l(U)=0\ GOTO\ N \\ IF\ l(U)=1\ GOTO\ A \\ IF\ P \nmid S\ GOTO\ N \\ IF\ l(U)=2\ GOTO\ M \\ K \longleftarrow \mu i \leq Long(Z)[l((Z)_i)+2 = l(U)] \\ GOTO\ C \end{array}\right\} (3)$$

paso de computación de X_{n+1}

$$\left.\begin{array}{ll} [M] & S \longleftarrow qt(S,P) \\ & GOTO\ N \\ [A] & S \longleftarrow S \cdot P \\ [N] & K \longleftarrow K+1 \\ & GOTO\ C \\ [F] & Y \longleftarrow (S)_1 \end{array}\right\} (4)$$

salida

Descripción del programa universal U_n

- **BLOQUE 1**: Iniciación del programa U_n.
  - En Z se almacena la información acerca del programa P:
  - En S el estado actual.
  - En K la siguiente instrucción que debe ejecutarse.

  (Si $K = k$ y $S = s$, el par (k, s) codifica una configuración)
- **BLOQUE 2**: Test de salida.

  Se llega a una configuración de parada de P si $K = Long(Z) + 1$ o bien si alguna instrucción condicional referencia a una etiqueta que no está a la izquierda de ninguna instrucción (en cuyo caso también se tiene que $K = Long(Z) + 1$).
- **BLOQUE 3**: Determina el tipo de instrucción que debe ejecutarse.
- **BLOQUE 4**: Calcula la configuración siguiente en la computación del programa codificado por X_{n+1} con entrada $(X_1, \dots, X_n)$.

Función universal

Teorema de la función Universal.
Existe una función $\mathcal{U} \in GCOMP^{(2)}$ tal que para cada $n \geq 1$ y cada $f \in GCOMP^{(n)}$, existe $e \in \mathbb{N}$ verificando que

$$\forall \vec{x} \in \mathbb{N}^n (f(\vec{x}) = \mathcal{U}([x_1, \dots, x_n], e))$$

- $\mathcal{U}$ es la función que calcula el programa universal U, diseñado como U_n pero cambiando el bloque de iniciación, $S \leftarrow \prod_{i=1}^{n} p_{2i}^{x_i}$ por

$$S \leftarrow \prod_{i=1}^{Long(x)} p_{2i}^{(x)_i}$$