Capítulo IV:
FUNCIONES RECURSIVAS

IV.1: MODELOS DE COMPUTACIÓN

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

Gracias, Alejandro

Modelo de Computación: definición matemática del concepto de *procedimiento mecánico*.

- *Sintaxis*.
- *Semántica*.

Modelos de computación orientado a:

- *Programas* (modelo `GOTO`).
- *Funciones* (funciones recursivas).
- *Máquinas* (máquinas de Turing).

Modos de computación:

- *Determinista*.
- *No-determinista*.
- *Secuencial*.
- *Paralelo*.

El modelo de computación GOTO (I)

Modelo orientado a programas M:

- Un conjunto de datos de entrada y salida, $\mathbb{D}$.
- Un conjunto de programas (sintaxis de M)
- Una descripción de cómo funciona cada programa (semántica de M).

A cada programa P en M y cada $k \geq 1$ se le asocia una función:

$$[\![P]\!]^{(k)} : \mathbb{D}^k - \longrightarrow \mathbb{D}$$

$[\![P]\!]^{(k)}(d)$= resultado de ejecutar P sobre el dato $d \in \mathbb{D}^k$.

Una función $f : \mathbb{D}^k - \longrightarrow \mathbb{D}$ es *computable en M* si existe un programa P en M tal que $[\![P]\!]^{(k)} = f$.

El modelo de computación GOTO (II)

Modelo secuencial–determinista con conjunto de datos $\mathbb{N}$.

1. **Variables**:
$$\begin{cases} \text{De entrada:} & X_1(=X), X_2, X_3, \ldots \\ \text{De salida:} & Y \\ \text{Auxiliares:} & Z_1(=Z), Z_2, Z_3, \ldots \end{cases}$$

2. **Etiquetas**: $A_1, B_1, C_1, D_1, E_1, A_2, B_2, C_2, D_2, E_2, \ldots$
   - Notación: $A = A_1$, $B = B_1$, $C = C_1$, $D = D_1$, $E = E_1$.

3. **Instrucciones**: Para cada variable V y cada etiqueta L:
   - **Incremento**: $V \longleftarrow V + 1$
   - **Decremento**: $V \longleftarrow V - 1$
   - **Condicional**: $IF\ V \neq 0\ GOTO\ L$
   - **Skip**: $V \longleftarrow V$
   - **Instrucciones etiquetadas**: Para cada etiqueta K:

$$[K]\ V \longleftarrow V + 1$$
$$[K]\ V \longleftarrow V - 1$$
$$[K]\ IF\ V \neq 0\ GOTO\ L$$
$$[K]\ V \longleftarrow V$$

Programas GOTO: Sintaxis

- Programa GOTO:
  - sucesión finita de instrucciones GOTO (etiquetadas o no) cuya última instrucción NO es $Y \longleftarrow Y$.
- Si $P = I_1, \ldots, I_n$ es un programa GOTO el número n se denomina longitud del programa y se denota por $|P|$.
- El Programa Vacío: consta de 0 instrucciones.
- $\mathtt{GOTO}_{\mathtt{P}}$: conjunto de los programas GOTO.
- Ejemplo:

$$P \equiv \begin{cases} [A] & X \longleftarrow X - 1 \\ & Y \longleftarrow Y + 1 \\ & \mathit{IF}\ X \neq 0\ \mathit{GOTO}\ A \end{cases}$$

Programas GOTO: Semántica (I)

Sea P un programa GOTO.

- VAR: conjunto de todas las variables de GOTO.
- $var(P)$: conjunto de todas las variables que aparecen en P.
- Un *estado* de P es una aplicación σ de $\mathtt{VAR} \to \mathbb{N}$.
  - Si $\sigma(V) = m$, notaremos $V = m$.
- Una *configuración* o *descripción instantánea* (d.i) de P, es un par $s = (i, \sigma)$ donde $1 \leq i \leq |P| + 1$ y σ es un estado de P.
  - Configuración inicial: $i = 1$, $\sigma(V) \neq 0$ para un número finito de variables de entrada y $\sigma(V) = 0$ para cada variable que no sea de entrada.
  - Configuración de parada: $i = |p| + 1$.
- Si $s = (i, \sigma)$, escribiremos $s(V)$ en lugar de $\sigma(V)$.

Programas GOTO: Semántica (II)

Sea $P = I_1, \ldots, I_n$ un programa GOTO.
Sea $s = (j, \sigma)$ una configuración de P que **no** es de parada.

- Diremos que la configuración $s' = (j', \sigma')$ es **la** *configuración siguiente* de s por P, y escribimos $s \vdash_p s'$, si:
  - Caso 1: $I_j \equiv V \longleftarrow V + 1$ y $s(V) = m$. Entonces $j' = j + 1$ y
    - σ' es σ sustituyendo $V = m$ por $V = m + 1$.
  - Caso 2: $I_j \equiv V \longleftarrow V - 1$ y $s(V) = m$. Entonces $j' = j + 1$ y
    - Si $m > 0$, σ' es σ sustituyendo $V = m$ por $V = m - 1$.
    - Si $m = 0$, $\sigma' = \sigma$
  - Caso 3: I_j es de tipo SKIP. Entonces $j' = j + 1$ y $\sigma' = \sigma$.
  - Caso 4: $I_j \equiv IF\ V \neq 0\ GOTO\ L$. Entonces $\sigma' = \sigma$ y
    - Si $s(V) = 0$, entonces $j' = j + 1$
    - Si $s(V) \neq 0$ y existe k tal que I_k es la primera instrucción etiquetada con L, entonces $j' = k$
    - Si $s(V) \neq 0$ y no existe tal k, $j' = |p| + 1$

Computación de un programa GOTO

Sea $P = I_1, \ldots, I_n$ un programa GOTO.

- Una *computación* $\mathcal{C}$ de P es una sucesión (finita o infinita) de configuraciones $s_1, \ldots, s_i, \ldots$ tal que:
  - $s_1 = (1, \sigma_1)$ es una configuración inicial;
  - Si s_i no es de parada, entonces $s_i \vdash_P s_{i+1}$.

En la anterior situación diremos que:

- $\mathcal{C}$ es **la** computación de P con dato de entrada $(a_1, \ldots, a_k)$, siendo $\sigma_1(X_j) = a_j$ $(1 \leq j \leq k)$ y $\sigma_1(X_j) = 0$, para $j \geq k$.

Además,

- Si $\mathcal{C} = s_1, \ldots, s_{i_0}$ es finita, diremos que
  - $\mathcal{C}$ es de parada: $P(a_1, \ldots, a_k) \downarrow$.
  - La configuración s_{i_0} es de parada.
  - $P(a_1, \ldots, a_k) = b$, siendo b el valor de Y en la configuración s_{i_0}.
- Si $\mathcal{C}$ es infinita, diremos que $\mathcal{C}$ no es de parada: $P(a_1, \ldots, a_k) \uparrow$.
- Las computaciones que **no** son de **parada no** proporcionan resultados.

Ejemplo

$$P \equiv \begin{array}{ll} [A] & IF\ X \neq 0\ GOTO\ B \\ & Z \longleftarrow Z + 1 \\ & IF\ Z \neq 0\ GOTO\ E \\ [B] & X \longleftarrow X - 1 \\ & Y \longleftarrow Y + 1 \\ & IF\ X \neq 0\ GOTO\ A \end{array}$$

Hallemos $P(3)$:

$$\begin{aligned} s_1 &= (1, \{X = 3,\ Y = 0,\ Z = 0\}) \\ s_2 &= (4, \{X = 3,\ Y = 0,\ Z = 0\}) \\ s_3 &= (5, \{X = 2,\ Y = 0,\ Z = 0\}) \\ s_4 &= (6, \{X = 2,\ Y = 1,\ Z = 0\}) \\ s_5 &= (1, \{X = 2,\ Y = 1,\ Z = 0\}) \\ s_6 &= (4, \{X = 2,\ Y = 1,\ Z = 0\}) \\ s_7 &= (5, \{X = 1,\ Y = 1,\ Z = 0\}) \\ s_8 &= (6, \{X = 1,\ Y = 2,\ Z = 0\}) \\ s_9 &= (1, \{X = 1,\ Y = 2,\ Z = 0\}) \\ s_{10} &= (4, \{X = 1,\ Y = 2,\ Z = 0\}) \\ s_{11} &= (5, \{X = 0,\ Y = 2,\ Z = 0\}) \\ s_{12} &= (6, \{X = 0,\ Y = 3,\ Z = 0\}) \\ s_{13} &= (7, \{X = 0,\ Y = 3,\ Z = 0\}) \end{aligned}$$

Funciones GOTO–computables

Sean P un programa GOTO y $k \geq 1$.

- Función de aridad k calculada por P:

$$[\![P]\!]^{(k)} : \mathbb{N}^k \dashrightarrow \mathbb{N}$$

definida por $[\![P]\!]^{(k)}(a_1, \ldots, a_k) = P(a_1, \ldots, a_k) =$ resultado de ejecutar P con dato de entrada $(a_1, \ldots, a_k)$.

Una función f de aridad $k \geq 1$ es GOTO–computable si existe un programa GOTO, P, tal que $f = [\![P]\!]^{(k)}$.

- G-COMP$^{(k)}$ es el conjunto de funciones GOTO computables de aridad k.
- G-COMP es el conjunto de funciones GOTO computables.

Ejemplos

- La función identidad $Id_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$ definida por $Id_N(x) = x$, para cada $x \in \mathbb{N}$, es GOTO–computable.

$$
\begin{array}{ll}
 & \mathit{IF}\ X \neq 0\ \mathit{GOTO}\ B \\
 & Z \longleftarrow Z + 1 \\
 & \mathit{IF}\ Z \neq 0\ \mathit{GOTO}\ E \\
[B] & X \longleftarrow X - 1 \\
 & Y \longleftarrow Y + 1 \\
 & \mathit{IF}\ X \neq 0\ \mathit{GOTO}\ B
\end{array}
$$

- La función vacía $f_{\emptyset}^{(k)}$ de aridad $k \geq 1$ definida por $f_{\emptyset}^{(k)}(x_1, \ldots, x_k) \uparrow$, para cada $(x_1, \ldots, x_k) \in \mathbb{N}^k$, es GOTO–computable:

$$
\begin{array}{ll}
[A] & Z \longleftarrow Z + 1 \\
 & \mathit{IF}\ Z \neq 0\ \mathit{GOTO}\ A
\end{array}
$$

- La función nula $\mathcal{O}^{(k)}$ de aridad $k \geq 1$ definida por $\mathcal{O}^{(k)}(x_1, \ldots, x_k) = 0$, para cada $(x_1, \ldots, x_k) \in \mathbb{N}^k$, es GOTO–computable.

Es calculada por cualquier programa que *pare* siempre y en el que no aparezca la variable de salida Y.

Macros GOTO (I)

Bloques de instrucciones GOTO que permiten realizar tareas específicas.

- El bloque de instrucciones:

$$Z \longleftarrow Z + 1$$
$$IF\ Z \neq 0\ GOTO\ L$$

  permite realizar un salto incondicional a la instrucción etiquetada por L (o terminar la ejecución del programa).

- El bloque de instrucciones:

$$[L] \quad V \longleftarrow V - 1$$
$$IF\ V \neq 0\ GOTO\ L$$

  pone a cero una variable V.

Macros GOTO (II)

Abreviaturas para *usar* tareas en la descripción de un programa: macros GOTO.

Cada macro GOTO realiza una tarea determinada: un programa GOTO que realiza esa tarea se denomina **una** expansión de la macro.

Ejemplos:

$$\underbrace{GOTO\ L}_{macro} \implies \underbrace{\left.\begin{array}{l} Z_k \longleftarrow Z_k + 1 \\ IF\ Z_k \neq 0\ GOTO\ L \end{array}\right\}}_{expansión}$$

$$\underbrace{V \longleftarrow 0}_{macro} \implies \underbrace{\left\{\begin{array}{ll} [K] & V \longleftarrow V - 1 \\ & IF\ V \neq 0\ GOTO\ K \end{array}\right.}_{expansión}$$

Notas importantes para la inserción de una macro en un programa P:

- En el primer ejemplo Z_k debe ser una variable que no aparezca en P.
- En el segundo, K debe ser una etiqueta que no aparezca en P.

Programa GOTO puro: carece de macros GOTO.

Programa GOTO impuro: posee macros GOTO.

Macros GOTO de asignación

El formato de una macro de asignación será: $V \longleftarrow V'$.

Tarea a realizar por esa macro:

- asignar a V el valor de V', conservando V' su valor.

Una expansión de la macro $V \longleftarrow V'$. es:

$$
\begin{array}{ll}
 & V \longleftarrow 0 \\
[A] & \mathit{IF}\ V' \neq 0\ \mathit{GOTO}\ B \\
 & \mathit{GOTO}\ C \\
[B] & V' \longleftarrow V' - 1 \\
 & V \longleftarrow V + 1 \\
 & Z \longleftarrow Z + 1 \\
 & \mathit{GOTO}\ A \\
[C] & \mathit{IF}\ Z \neq 0\ \mathit{GOTO}\ D \\
 & \mathit{GOTO}\ L \\
[D] & Z \longleftarrow Z - 1 \\
 & V' \longleftarrow V' + 1 \\
 & \mathit{GOTO}\ C
\end{array}
$$

Copia V' en V y en Z y Pone V' a cero

Copia Z en V' Pone Z a cero

Macros GOTO condicionales

Si $\theta(V_1, \ldots, V_k)$ es un predicado GOTO computable de aridad k:

$$\left.\begin{array}{c} \textit{expansión} \\ Z \longleftarrow P(V_1, \ldots, V_k) \\ \textit{IF } Z \neq 0 \textit{ GOTO } L \end{array}\right\} \quad \begin{array}{c} \textit{macro} \\ \rightsquigarrow \textit{IF } \theta(V_1, \ldots, V_k) \neq 0 \textit{ GOTO } L \end{array}$$

- Ejemplo: $\textit{IF } V = 0 \textit{ GOTO } L$ es una macro GOTO

$$V = 0 \text{ es el predicado} \equiv \theta(x) = \begin{cases} 1 & \texttt{si} \quad x = 0 \\ 0 & \texttt{si} \quad x \neq 0 \end{cases}$$

donde $\theta(x)$ es G-computable: $\begin{cases} \textit{IF } X \neq 0 \textit{ GOTO } E \\ Y \longleftarrow Y + 1 \end{cases}$

Ejemplos de uso de las macros (I)

La *función suma*: $Sum : \mathbb{N}^2 \longrightarrow \mathbb{N}$; $Sum(x, y) = x + y$

$$P_+ \left\{ \begin{array}{ll} & Y \longleftarrow X_1 \\ & Z \longleftarrow X_2 \\ [B] & IF\ Z \neq 0\ GOTO\ A \\ & GOTO\ E \\ [A] & Z \longleftarrow Z - 1 \\ & Y \longleftarrow Y + 1 \\ & GOTO\ B \end{array} \right.$$

Ejercicio: Prueba de la corrección para la suma.

$$(\forall x, y \in \mathbb{N})([\![P_+]\!]^{(2)}(x, y) = x + y)$$

(por inducción débil sobre la segunda variable).

Ejemplos de uso de las macros (II)

La *función producto*: $Prod : \mathbb{N}^2 \longrightarrow \mathbb{N}$; $Prod(x, y) = x \cdot y$

$$p_{\times} \begin{cases} & Z_2 \longleftarrow X_2 \\ [B] & IF\ Z_2 \neq 0\ GOTO\ A \\ & GOTO\ E \\ [A] & Z_2 \longleftarrow Z_2 - 1 \\ & Z_1 \longleftarrow X_1 + Y \\ & Y \longleftarrow Z_1 \\ & GOTO\ B \end{cases}$$

Nota: $Z_1 \longleftarrow X_1 + Y$ es, a su vez, una macro que ejecuta el programa *Suma* y asigna el resultado a la variable Z_1.

Inserción de una macro en un programa GOTO

Para insertar una macro en un programa GOTO:

- Se considera una expansión de la macro.
- Las variables y etiquetas que aparecen en la expansión se renombran a fin de que sean distintas de las usadas en el programa.
- Si existe alguna instrucción del programa tras el lugar donde se inserte la macro:
  - Las etiquetas de salida de la expansión se renombran por la etiqueta de la primera instrucción que sigue a la expansión, si existe esa etiqueta.
  - Caso contrario se le asigna una nueva etiqueta L y la siguiente instrucción se etiqueta a la izquierda con L.