Tema 3
Universalidad

Modelos de Computación

Modelos de Computación - Universidad de Cádiz

Contenidos

1. Codificación
2. Codificación de Programas
3. El Problema de la Parada
4. Universalidad
5. Conjuntos Recursivamente Enumerables y Recursivos
6. Autoreferencia

Codificación

Definición (Codificación de Programas)

$$\#_{\mathscr{P}} : Prog \rightarrow \mathbb{N}.$$

Codificación

Definición (Codificación de Variables)

Ordenamiento de la lista de variables: $y, x_1, z_1, x_2, z_2, x_3, z_3 \ldots$

$$\#_V : \mathscr{V}ar \to \mathbb{N},$$

$$\#_V(V) = \text{posición de V en la lista ordenada de variables.}$$

Ejemplo (Codificación de Variables)

$$\#_V(y) = 1, \qquad \#_V(x_2) = 4.$$

Codificación

Definición (Codificación de Etiquetas)

Ordenamiento de la lista de etiquetas: $A, B, C, D, E, A_2, B_2, C_2, D_2, E_2 \ldots$

$$\#_L : \mathscr{L}ab \to \mathbb{N},$$

$$\#_L(L) = \text{posición de L en la lista ordenada de etiquetas.}$$

Ejemplo (Codificación de Etiquetas)

$$\#_L(C_1) = 3, \qquad \#_L(B_2) = 7.$$

Codificación

Definición (Codificación de Instrucciones)

$$\#_I : I \to \mathbb{N}, \qquad \#_I(I) = \langle a, \langle b, c \rangle\rangle.$$

Reglas de Codificación (1)

- Elementos a considerar:
  1. La variable (c).
  2. Tipología (b).
  3. Etiquetado (a).
- Si I está etiquetada, $a = \#_L(L)$. Si no lo está, $a = 0$.
- Si V es la variable que aparece en I, entronces $c = \#_V(V) - 1$.

Reglas de Codificación (2)

- Si I es del tipo de instrucción:

Tipo	Valor de b
`V ← V`	0
`V ← V + 1`	1
`V ← V - 1`	2
`IF V ≠ GOTO L`	$2 + \#_L(L)$

Codificación

Ejemplo (Codificación de Instrucciones)

Instrucción	a	b	c	$\#_I$
$\texttt{Y} \leftarrow \texttt{Y}$	0	0	0	$\langle 0, \langle 0, 0 \rangle \rangle$
$\texttt{X}_1 \leftarrow \texttt{X}_1 + 1$	0	1	1	$\langle 0, \langle 1, 1 \rangle \rangle$
$[\texttt{A}_2]\ \texttt{X}_1 \leftarrow \texttt{X}_1 + 1$	6	1	1	$\langle 6, \langle 1, 1 \rangle \rangle$
$\texttt{IF}\ \texttt{Z}_3 \neq 0\ \texttt{GOTO}\ \texttt{A}_2$	0	8	6	$\langle 0, \langle 8, 6 \rangle \rangle$
$[\texttt{B}_1]\ \texttt{IF}\ \texttt{Z}_3 \neq 0\ \texttt{GOTO}\ \texttt{A}_2$	2	8	6	$\langle 2, \langle 8, 6 \rangle \rangle$

Codificación

Definición (Decodificación de Instrucciones)

Dado un $z \in \mathbb{N}$, se decodifica como sigue:

- Si $l(z) = 0$, entonces no hay etiqueta. En otro caso, la etiqueta es la $l(z)$-ésima en la lista de etiquetas.
- La variable codificada es la $r(r(z)) + 1$-ésima en la lista de variables.
- Dado $b = l(r(z))$,
  - Si $b = 0$, entonces la instrucción es de la forma `V` $\leftarrow$ `V`.
  - Si $b = 1$, entonces la instrucción es de la forma `V` $\leftarrow$ `V + 1`.
  - Si $b = 2$, entonces la instrucción es de la forma `V` $\leftarrow$ `V - 1`.
  - Si $b > 2$, entonces la instrucción es de la forma `IF V` $\neq$ `0 GOTO L`, y `L` es la $(b-2)$-ésima etiqueta en la lista ordenada.

Codificación de Programas

Definición

$$\#_P : P \to \mathbb{N}, \qquad \#_P(\mathscr{P}) = \text{Código de } \mathscr{P}.$$

Dado $P \in \mathscr{P}$, si $I_1, I_2, \ldots, I_n$ son las instrucciones de P, entonces

$$\#_P(P) = [\#_I(I_1), \#_I(I_2), \ldots, \#_I(I_n)].$$

Criterio

No se permite que $\mathtt{Y} \leftarrow \mathtt{Y}$ sea la última instrución de un programa.

El Problema de la Parada

Definición

Se define el predicado *Para* : $\mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ dado por

$$Para(x, y) \Leftrightarrow \text{el programa n}^{\text{o}}\ y \text{ para y ofrece un resultado sobre el input } x.$$

Teorema

El predicado $Para(x, y)$ no es computable.

Universalidad

Definición (Funciones Universales)

Para cada $n > 0$ se define la función universal dada por la siguiente expresión:

$$\phi^{(n)}(x_1, x_2, ..., x_n, y) = \psi_P^{(n)}(x_1, x_2, ..., x_n),$$

siendo $y = \#_P(P)$.
Cambiar el parámetro y es programar la función universal.

Teorema (de la Universalidad)

Para cada $n > 0$, la función $\phi^{(n)}(x_1, x_2, ..., x_n, y)$ es parcialmente computable.

Intérprete

Almacenamiento del Intérprete

- Memoria del programa interpretado:

$$\mathtt{S} \leftarrow \prod_{i=1}^{n} (\mathtt{P}_{2i})^{x_i}.$$

- Programa a interpretar codificado:

$$\mathtt{Z} \leftarrow \mathtt{X}_{n+1}\mathtt{+1}.$$

- Contador de programa:

$$\mathtt{K} \leftarrow \mathtt{1}.$$

Intérprete

Criterio de Parada del Intérprete

$$\texttt{[C] IF R = Lt(Z)+1} \vee \texttt{R=0 GOTO F.}$$

Determinando la Instrucción Actual

$$\texttt{U} \leftarrow \texttt{r((Z)}_k\texttt{)},$$
$$\texttt{P} \leftarrow \texttt{P}_{r(U)+1}.$$

Intérprete

Filtro de Codificación

- Instrucción tipo dummy:

  `IF l(U)=0 GOTO N.`

- Instrucción tipo incremento:

  `IF l(U)=1 GOTO A.`

- Instrucción tipo `IF V` $\neq$ `0 GOTO L`:

  `IF` $\neg$`(P|S) GOTO N.`

- Instrucción tipo decremento:

  `IF l(U)=2 GOTO M.`

Intérprete

Núcleo del Intérprete

```
[M]   S ← ⌊S/P⌋
      GOTO N
[A]   S ← S·P
[N]   K ← K+1
      GOTO C
```

Intérprete Completo

$$
\begin{array}{lll}
 & \mathtt{Z} & \leftarrow \ \mathtt{X}_{n+1}\mathtt{+1} \\
 & \mathtt{S} & \leftarrow \ \prod_{i=1}^{n} (\mathtt{P}_{2i})^{\mathtt{X}_i} \\
 & \mathtt{K} & \leftarrow \ \mathtt{1} \\
[\mathtt{C}] & \mathtt{IF\ R\ =\ Lt(Z)+1} & \vee \ \mathtt{R=0\ GOTO\ F} \\
 & \mathtt{U} & \leftarrow \ \mathtt{r((Z)}_k\mathtt{)} \\
 & \mathtt{P} & \leftarrow \ \mathtt{P}_{r(U)+1} \\
 & \mathtt{IF\ l(U)=0\ GOTO\ N} & \\
 & \mathtt{IF\ l(U)=1\ GOTO\ A} & \\
 & \mathtt{IF\ \neg(P|S)\ GOTO\ N} & \\
 & \mathtt{IF\ l(U)=2\ GOTO\ M} & \\
 & \mathtt{K} & \leftarrow \ \min_{i \leqslant Lt(z)} [\mathtt{l((Z)}_i\mathtt{)+2=l(u)}] \\
 & \mathtt{GOTO\ C} & \\
[\mathtt{M}] & \mathtt{S} & \leftarrow \ \lfloor \mathtt{S/P} \rfloor \\
 & \mathtt{GOTO\ N} & \\
[\mathtt{A}] & \mathtt{S} & \leftarrow \ \mathtt{S \cdot P} \\
[\mathtt{N}] & \mathtt{K} & \leftarrow \ \mathtt{K+1} \\
 & \mathtt{GOTO\ C} & \\
[\mathtt{F}] & \mathtt{Y} & \leftarrow \ \mathtt{(S)}_1
\end{array}
$$

Predicado Contador

Definición (Predicado Contador)

Para cada $n > 0$, se define

$$STP^{(n)}(x_1, x_2, ..., x_n, y, t) \Leftrightarrow \text{el programa número } y \text{ para sobre el input } (x_1, x_2, ..., x_n) \text{ en } t \text{ o menos pasos de cálculo.}$$

Proposición

Para cada $n > 0$, $STP^{(n)}(x_1, x_2, ..., x_n, y, t)$ *es computable.*

Conjuntos Recursivamente Enumerables y Recursivos

Conjuntos Recursivamente Enumerables y Recursivos

$$B \subseteq \mathbb{N} \begin{cases} \text{Recursivos} & \begin{cases} P_B \text{ es computable.} \\ \text{La pertenencia es decidible.} \end{cases} \\ \\ \text{Recursivos Enumerables} & \begin{cases} f(x) \text{ es parcialmente computable.} \\ \text{La pertenencia es semidecidible.} \end{cases} \end{cases}$$

Conjuntos Recursivamente Enumerables y Recursivos

Definición (Función Característica)

Sea $B \subseteq \mathbb{N}$, se define la función caractetística P_B como

$$P_B : \mathbb{N} \to \{0,1\}, \qquad P_B(x) = \begin{cases} 1 & \text{si } x \in B \\ 0 & \text{en otro caso.} \end{cases}$$

Ejemplo

Sea $B \subseteq \mathbb{N}$ y dado por $B = \{x \in \mathbb{N} : x \equiv 0 mod 7\}$.

Conjuntos Recursivamente Enumerables y Recursivos

Definición (Conjunto Recursivamente Enumerable)

Dado $B \subseteq \mathbb{N}$, se dice que B es recursivamente enumerable si existe una función parcialmente computable $f : \mathbb{N} \to \mathbb{N}$ tal que

$$B = \{x \in \mathbb{N} : f(x) \downarrow\}.$$

Proposición

Si $B \subseteq \mathbb{N}$ es recursivo, entonces es recursivo enumerable.

Teorema

Dado $B \subseteq \mathbb{N}$, B es recursivo si y solo si B y $\bar{B}$ son recursivos enumerables.

Teorema

Sean $B, C \subseteq \mathbb{N}$, ambos recursivos enumerables. Entonces $B \cup C$, $B \cap C$ son también recursivos enumerables.

Autoreferencia

Definición

Para cada $n > 0$, sea $W_n = \{x \in \mathbb{N} : \phi(x, n) \downarrow\}$.

Teorema (de Enumeración)

$B \subseteq \mathbb{N}$ es recursivo enumerable $\Leftrightarrow$ existe $n \in \mathbb{N}$ tal que $B = W_n$.

Autoreferencia

Definición

Se define

$$k = \{n \in \mathbb{N} : \phi(n, n) \downarrow\}.$$

Teorema

k es recursivo enumerable pero no es recursivo.