

Turing vence a von Neumann

Agustín Curto

FaMAF - UNC

2017

Introduciendo notación

Notación

Dados $x_1, \dots, x_n \in \omega$ y $\alpha_1, \dots, \alpha_m \in \Sigma^*$, con $n, m \in \omega$, usaremos:

$$\|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\|$$

para denotar el estado

$$((x_1, \dots, x_n, 0, \dots), (\alpha_1, \dots, \alpha_m, \varepsilon, \dots))$$

Nótese que por ejemplo:

$$\|x\| = ((x, 0, \dots), (\varepsilon, \dots)) \quad \text{Para } n = 1, m = 0$$

$$\|\diamond\| = ((0, \dots), (\varepsilon, \dots)) \quad \text{Para } n = m = 0$$

Además es claro que:

$$\|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\| = \|x_1, \dots, x_n, \overbrace{0, \dots, 0}^i, \alpha_1, \dots, \alpha_m, \overbrace{\varepsilon, \dots, \varepsilon}^j\|$$

cualesquiera sean $i, j \in \omega$.

Toda función Σ -computable es Σ -Turing computable

Probaremos

Si f es una función Σ -mixta que es computada por un programa $\mathcal{P} \in \text{Pro}^\Sigma$, entonces existe una máquina de Turing determinística con unit M la cual computa a f .

Definición

Dado $\mathcal{P} \in \text{Pro}^\Sigma$, definamos:

$$N(\mathcal{P}) = \text{menor } k \in \mathbb{N} \text{ tal que las variables que ocurren en } \mathcal{P} \\ \text{están todas en la lista } N1, \dots, N\bar{k}, P1, \dots, P\bar{k}$$

Ejemplo: Sea $\Sigma = \{\&, \#\}$, si \mathcal{P} es el siguiente programa:

```
L1  N4  $\leftarrow$  N4 + 1  
    P1  $\leftarrow$  P1.&  
    IF N1  $\neq$  0 GOTO L1
```

entonces tenemos $N(\mathcal{P}) = 4$

Sea \mathcal{P} un programa y sea k fijo y $k \geq N(\mathcal{P})$. Describiremos como puede construirse una máquina de Turing la cual simulará a \mathcal{P} . La construcción de la máquina simuladora dependerá de \mathcal{P} y de k .

Nótese que cuando \mathcal{P} se corre desde algún estado de la forma

$$\|x_1, \dots, x_k, \alpha_1, \dots, \alpha_k\|$$

los sucesivos estados por los que va pasando son todos de la forma

$$\|y_1, \dots, y_k, \beta_1, \dots, \beta_k\|$$

es decir, en todos ellos las variables con índice mayor que k valen 0 o ε .

La razón es simple, ya que en \mathcal{P} no figuran las variables

$$\overline{Nk+1}, \overline{Nk+2}, \dots$$

$$\overline{Pk+1}, \overline{Pk+2}, \dots$$

estas variables quedan con valores 0 y ε , respectivamente a lo largo de toda la computación.

Necesitaremos tener alguna manera de representar en la cinta los diferentes estados por los cuales se va pasando, a medida que corremos a \mathcal{P} . Esto lo haremos de la siguiente forma, al estado

$$\|x_1, \dots, x_k, \alpha_1, \dots, \alpha_k\|$$

lo representaremos en la cinta de la siguiente manera

$$B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k B B B B \dots$$

Ejemplo: consideremos el programa \mathcal{P} mostrado recién y fijemos $k = 6$, entonces al estado

$$\|3, 2, 5, 0, 4, 2, \&, \&\&, \varepsilon, \# \&, \#, \#\#\#\|$$

lo representaremos en la cinta de la siguiente manera

$$B \mid \mid \mid B \mid \mid B \mid \mid \mid \mid B B \mid \mid \mid \mid B \mid \mid B \& B \& \& B B \# \& B \# B \# \# \# B B B B B \dots$$

Definición

A lo que queda entre dos blancos consecutivos, es decir, que no hay ningún blanco entre ellos, lo llamaremos *bloque*.

Ejemplo: en la cinta anterior tenemos que los primeros 12 bloques son

||| || |||| ε |||| || & && ε #& # ###

luego, los bloques siguientes, son todos iguales a ε .

Observación

Esta forma de representación de estados en la cinta depende del k elegido, es decir, si tomáramos otro k , por ejemplo $k = 9$, entonces el estado anterior se representaría de otra forma en la cinta.

- Armaremos la máquina simuladora como concatenación de máquinas. Para esto, a continuación describiremos, para los distintos tipos de instrucciones posibles de \mathcal{P} , sus respectivas máquinas asociadas.
- Asumiremos que en \mathcal{P} no hay instrucciones de la forma $\text{GOTO } L\bar{m}$, ni de la forma $L\bar{n} \text{ GOTO } L\bar{m}$.
- En esta etapa solo describiremos que propiedades tendrá que tener cada máquina simuladora de cada tipo posible de instrucción, y más adelante mostraremos como pueden ser construídas efectivamente dichas máquinas.
- Todas las máquinas descriptas tendrán:
 - \perp como unit
 - B como blanco
 - Σ como su alfabeto terminal
 - su alfabeto mayor será $\Gamma = \Sigma \cup \{B, \perp\} \cup \{\tilde{a} : a \in \Sigma \cup \{\perp\}\}$.
 - uno o dos estados finales con la siguiente propiedad:

Si q es un estado final $\Rightarrow \delta(q, \sigma) = \emptyset$, para cada $\sigma \in \Gamma$

Instrucción $N\bar{i} \leftarrow N\bar{i} + 1$

Para $1 \leq i \leq k$, sea $M_{i,k}^+$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k \\ \uparrow \\ q_0 \end{array} \stackrel{*}{\vdash} \begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_i-1} B \mid^{x_i+1} B \mid^{x_{i+1}} \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k \\ \uparrow \\ q_f \end{array}$$

Instrucción $N\bar{i} \leftarrow N\bar{i} - 1$

Para $1 \leq i \leq k$, sea $M_{i,k}^-$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k \\ \uparrow \\ q_0 \end{array} \stackrel{*}{\vdash} \begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_i-1} B \mid^{x_i-1} B \mid^{x_{i+1}} \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k \\ \uparrow \\ q_f \end{array}$$

Instrucción $P\bar{i} \leftarrow P\bar{i}.a$

Para $1 \leq i \leq k$ y $a \in \Sigma$, sea $M_{i,k}^a$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k & \stackrel{*}{\vdash} & B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_{i-1} B \alpha_i a B \alpha_{i+1} B \dots B \alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

Instrucción $P\bar{i} \leftarrow \curvearrowright P\bar{i}$

Para $1 \leq i \leq k$, sea $M_{i,k}^{\curvearrowright}$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k & \stackrel{*}{\vdash} & B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_{i-1} B^{\curvearrowright} \alpha_i B \alpha_{i+1} B \dots B \alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

Instrucción $N\bar{i} \leftarrow N\bar{j}$

Para $1 \leq i, j \leq k$, sea $M_{i \leftarrow j}^{\#,k}$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_k \\ \uparrow \\ q_0 \end{array} \vdash^* \begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_i-1} B \mid^{x_j} B \mid^{x_{j+1}} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_k \\ \uparrow \\ q_f \end{array}$$

Instrucción $P\bar{i} \leftarrow P\bar{j}$

Para $1 \leq i, j \leq k$, sea $M_{i \leftarrow j}^{*,k}$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_k \\ \uparrow \\ q_0 \end{array} \vdash^* \begin{array}{c} B \mid^{x_1} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_{i-1} B\alpha_j B\alpha_{i+1} B \dots B\alpha_k \\ \uparrow \\ q_f \end{array}$$

Instrucción $N\bar{i} \leftarrow 0$

Para $1 \leq i \leq k$, sea $M_{i \leftarrow 0}^k$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k & \vdash^* & B \mid^{x_1} B \dots B \mid^{x_{i-1}} B B \mid^{x_{i+1}} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

Instrucción $P\bar{i} \leftarrow \varepsilon$

Para $1 \leq i \leq k$, sea $M_{i \leftarrow \varepsilon}^k$ una máquina con estado inicial q_0 y único estado final q_f tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k & \vdash^* & B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_{i-1} B B \alpha_{i+1} B \dots B \alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

Instrucción SKIP

Sea

$$M_{\text{SKIP}} = (\{q_0, q_f\}, \Gamma, \Sigma, \delta, q_0, B, \mid, \{q_f\})$$

con $\delta(q_0, B) = \{(q_f, B, K)\}$ y $\delta = \emptyset$ en cualquier otro caso.

Instrucción IF $N\bar{j} \neq 0$ GOTO $L\bar{m}$

Para $1 \leq j \leq k$, sea $IF_{j,k}$ una máquina con estado inicial q_0 y dos estados finales q_{si} y q_{no} tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

- Si $x_j \neq 0$, entonces

$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_k & \overset{*}{\vdash} & B \mid^{x_1} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_{si} \end{array}$$

- Si $x_j = 0$, entonces

$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_k & \overset{*}{\vdash} & B \mid^{x_1} B \dots B \mid^{x_k} B\alpha_1 B \dots B\alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_{no} \end{array}$$

Instrucción IF $P_j^{\bar{}}$ BEGINS a GOTO $L_{\bar{m}}$

Para $1 \leq j \leq k$ y $a \in \Sigma$, sea $IF_{j,k}^a$ una máquina con estado inicial q_0 y dos estados finales q_{si} y q_{no} tal que cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$

- Si α_j comienza con a , entonces

$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k & \stackrel{*}{\vdash} & B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_{si} \end{array}$$

- Caso contrario

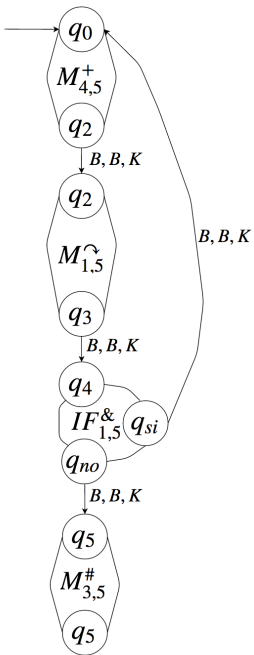
$$\begin{array}{ccc} B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k & \stackrel{*}{\vdash} & B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_{no} \end{array}$$

Example

Sea $\Sigma = \{\&, \#\}$ y sea \mathcal{P} el siguiente programa:

```
L3  N4  $\leftarrow$  N4 + 1  
    P1  $\leftarrow$   $\neg$ P1  
    IF P1 BEGINS & GOTO L3  
    P3  $\leftarrow$  P3.#
```

Tomemos $k = 5$, es claro que $k \geq N(\mathcal{P}) = 4$. A la máquina que simulará a \mathcal{P} respecto de k , la llamaremos M_{sim} y será la siguiente:



Veamos con un ejemplo como M_{sim} simula a \mathcal{P} . Supongamos que corremos \mathcal{P} desde el estado

$\|2, 1, 0, 5, 3, \# \& \# \#, \varepsilon, \& \&, \# \&, \# \|$

Tendremos entonces la siguiente sucesión de descripciones instantáneas:

(1, $\|2, 1, 0, 5, 3, \# \& \# \#, \varepsilon, \& \&, \# \&, \# \|$)

(2, $\|2, 1, 0, 6, 3, \# \& \# \#, \varepsilon, \& \&, \# \&, \# \|$)

(3, $\|2, 1, 0, 6, 3, \& \# \#, \varepsilon, \& \&, \# \&, \# \|$)

(1, $\|2, 1, 0, 6, 3, \& \# \#, \varepsilon, \& \&, \# \&, \# \|$)

(2, $\|2, 1, 0, 7, 3, \& \# \#, \varepsilon, \& \&, \# \&, \# \|$)

(3, $\|2, 1, 0, 7, 3, \# \#, \varepsilon, \& \&, \# \&, \# \|$)

(4, $\|2, 1, 0, 7, 3, \# \#, \varepsilon, \& \&, \# \&, \# \|$)

(5, $\|2, 1, 0, 7, 3, \# \#, \varepsilon, \& \& \#, \# \&, \# \|$)

Si hacemos funcionar a M_{sim} desde:

$$q_0 B \mid^2 B \mid BB \mid^5 B \mid^3 B \# \& \# \# BB \& \& B \# \& B \# B$$

obtendremos una sucesión de descripciones instantáneas dentro de las cuales estará la siguiente subsucesión

$$q_0 B \mid^2 B \mid BB \mid^5 B \mid^3 B \# \& \# \# BB \& \& B \# \& B \# B$$

$$q_1 B \mid^2 B \mid BB \mid^6 B \mid^3 B \# \& \# \# BB \& \& B \# \& B \# B$$

$$q_2 B \mid^2 B \mid BB \mid^6 B \mid^3 B \# \& \# \# BB \& \& B \# \& B \# B$$

$$q_3 B \mid^2 B \mid BB \mid^6 B \mid^3 B \& \# \# BB \& \& B \# \& B \# B$$

$$q_4 B \mid^2 B \mid BB \mid^6 B \mid^3 B \& \# \# BB \& \& B \# \& B \# B$$

$$q_{si} B \mid^2 B \mid BB \mid^6 B \mid^3 B \& \# \# BB \& \& B \# \& B \# B$$

$$q_0 B \mid^2 B \mid BB \mid^6 B \mid^3 B \& \# \# BB \& \& B \# \& B \# B$$

$q_1 B \mid^2 B \mid BB \mid^7 B \mid^3 B \& \# \# BB \& \& B \# \& B \# B$

$q_2 B \mid^2 B \mid BB \mid^7 B \mid^3 B \& \# \# BB \& \& B \# \& B \# B$

$q_3 B \mid^2 B \mid BB \mid^7 B \mid^3 B \# \# BB \& \& B \# \& B \# B$

$q_4 B \mid^2 B \mid BB \mid^7 B \mid^3 B \# \# BB \& \& B \# \& B \# B$

$q_{no} B \mid^2 B \mid BB \mid^7 B \mid^3 B \# \# BB \& \& B \# \& B \# B$

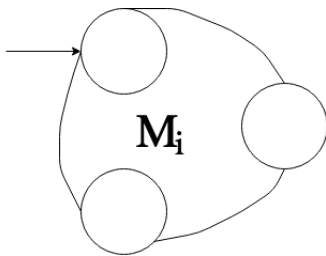
$q_5 B \mid^2 B \mid BB \mid^7 B \mid^3 B \# \# BB \& \& B \# \& B \# B$

$q_6 B \mid^2 B \mid BB \mid^7 B \mid^3 B \# \# BB \& \& \# B \# \& B \# B$

Supongamos que $\mathcal{P} = I_1, \dots, I_n$. Para cada $i = 1, \dots, n$, llamaremos M_i a la máquina que simulará el efecto que produce la instrucción I_i , es decir tomemos:

- $M_i = M_{j,k}^+$, si $Bas(I_i) = N\bar{j} \leftarrow N\bar{j} + 1$
- $M_i = M_{j,k}^-$, si $Bas(I_i) = N\bar{j} \leftarrow N\bar{j} - 1$
- $M_i = M_{j,k}^a$, si $Bas(I_i) = P\bar{j} \leftarrow P\bar{j}.a$
- $M_i = M_{j,k}^\cap$, si $Bas(I_i) = P\bar{j} \leftarrow \cap P\bar{j}$
- $M_i = M_{j \leftarrow m}^{\#,k}$, si $Bas(I_i) = N\bar{j} \leftarrow N\bar{m}$
- $M_i = M_{j \leftarrow m}^{*,k}$, si $Bas(I_i) = P\bar{j} \leftarrow P\bar{m}$
- $M_i = M_{j \leftarrow 0}^k$, si $Bas(I_i) = N\bar{j} \leftarrow 0$
- $M_i = M_{j \leftarrow \varepsilon}^k$, si $Bas(I_i) = P\bar{j} \leftarrow \varepsilon$
- $M_i = M_{\text{SKIP}}$, si $Bas(I_i) = \text{SKIP}$
- $M_i = IF_{j,k}$, si $Bas(I_i) = \text{IF } N\bar{j} \neq 0 \text{ GOTO } L\bar{m}$, para algún m
- $M_i = IF_{j,k}^a$, si $Bas(I_i) = \text{IF } P\bar{j} \text{ BEGINS } a \text{ GOTO } L\bar{m}$, para algún m

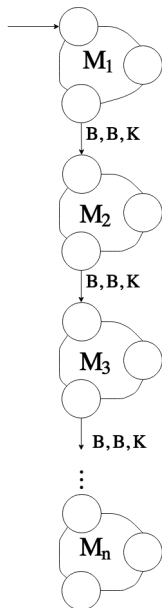
Ya que la máquina M_i puede tener uno o dos estados finales, la representaremos de la siguiente manera:



- Si M_i tiene **un solo estado final**, este está representado por el círculo de abajo a la izquierda
- Si M_i tiene **dos estados finales**, el estado final de la derecha corresponde al estado q_{si} y el de la izquierda abajo al estado q_{no} .

Para armar la máquina que simulará a \mathcal{P} hacemos lo siguiente:

1) Primero unimos las máquinas M_1, \dots, M_n de la siguiente manera



- 2) Luego para cada i tal que $Bas(I_i)$ es de la forma α GOTO $L\bar{m}$, ligamos con una flecha de la forma

$$\xrightarrow{B, B, K}$$

el estado final q_{si} de la M_i con el estado inicial de la M_h , donde h es tal que I_h es la primer instrucción que tiene label $L\bar{m}$.

Lema

Sea $\mathcal{P} \in \text{Pro}^\Sigma$ y sea $k \geq N(\mathcal{P})$. Supongamos que en \mathcal{P} no hay instrucciones de la forma GOTO $L\bar{m}$ ni de la forma $L\bar{n}$ GOTO $L\bar{m}$. Sean:

- Para cada $a \in \Sigma \cup \{|\}$, sea \tilde{a} un nuevo símbolo
- $\Gamma = \Sigma \cup \{B, |\} \cup \{\tilde{a} : a \in \Sigma \cup \{|\}\}$

entonces existe una máquina de Turing determinística con unit $M = (Q, \Gamma, \Sigma, \delta, q_0, B, |, \{q_f\})$, la cual satisface:

- 1) $\delta(q_f, \sigma) = \emptyset$, para cada $\sigma \in \Gamma$.
- 2) Cualesquiera sean $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$, el programa \mathcal{P} se detiene partiendo del estado

$$\|x_1, \dots, x_k, \alpha_1, \dots, \alpha_k\|$$

si y solo si M se detiene partiendo de la descripción instantánea

$$[q_0 B |^{x_1} B \dots B |^{x_k} B \alpha_1 B \dots B \alpha_k B]$$

3) Si $x_1, \dots, x_k \in \omega$ y $\alpha_1, \dots, \alpha_k \in \Sigma^*$ son tales que \mathcal{P} se detiene partiendo del estado

$$\|x_1, \dots, x_k, \alpha_1, \dots, \alpha_k\|$$

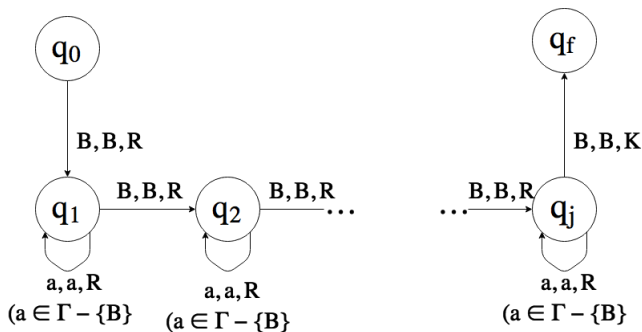
y llega al estado

$$\|y_1, \dots, y_k, \beta_1, \dots, \beta_k\|$$

entonces

$$[q_0 B \mid^{x_1} B \dots B \mid^{x_k} B \alpha_1 B \dots B \alpha_k B] \stackrel{*}{\vdash}_M [q_f B \mid^{y_1} B \dots B \mid^{y_k} B \beta_1 B \dots B \beta_k B]$$

Para cada $j \geq 1$, sea D_j la máquina siguiente:



Nótese que:

$$\begin{array}{ccc}
 \alpha B \beta_1 B \beta_2 B \dots B \beta_j B \gamma & \stackrel{*}{\vdash} & \alpha B \beta_1 B \beta_2 B \dots B \beta_j B \gamma \\
 \uparrow & & \uparrow \\
 q_0 & & q_f
 \end{array}$$

siempre que $\alpha, \gamma \in \Gamma^*$, $\beta_1, \dots, \beta_j \in (\Gamma - \{B\})^*$.

I_j cumplirá que:

$$\begin{array}{ccc} \alpha B \beta_j B \dots B \beta_2 B \beta_1 B \gamma & \overset{*}{\vdash} & \alpha B \beta_j B \dots B \beta_2 B \beta_1 B \gamma \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

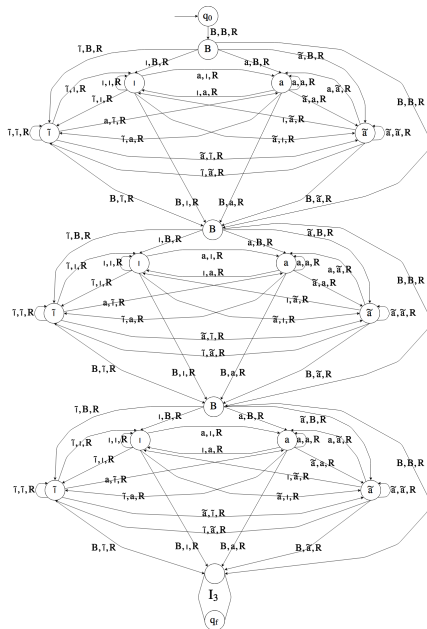
siempre que $\alpha, \gamma \in \Gamma^*$, $\beta_1, \dots, \beta_j \in (\Gamma - \{B\})^*$. Dejamos al lector la manufactura de esta máquina.

Para $j \geq 1$, sea TD_j una máquina con un solo estado final q_f y tal que:

$$\begin{array}{ccc} \alpha B \gamma & \overset{*}{\vdash} & \alpha B B \gamma \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

cada vez que $\alpha, \gamma \in \Gamma^*$ y γ tiene exactamente j ocurrencias de B .

Ejemplo: Sea $\Sigma = \{a\}$ podemos tomar TD_3 igual a la siguiente máquina



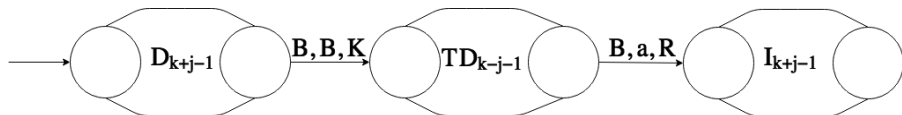
Análogamente, para $j \geq 1$, sea Tl_j una máquina tal que

$$\begin{array}{ccc} \alpha B \sigma \gamma & \overset{*}{\vdash} & \alpha B \gamma \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

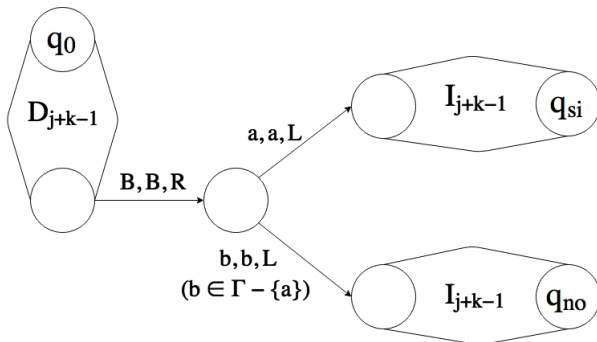
cada vez que $\alpha \in \Gamma^*$, $\sigma \in \Gamma$ y γ tiene exactamente j ocurrencias de B .
Dejamos al lector la construcción de, por ejemplo, Tl_3 para $\Sigma = \{\&, \#\}$.

Teniendo las máquinas auxiliares antes definidas podemos combinarlas para obtener las máquinas simuladoras de instrucciones.

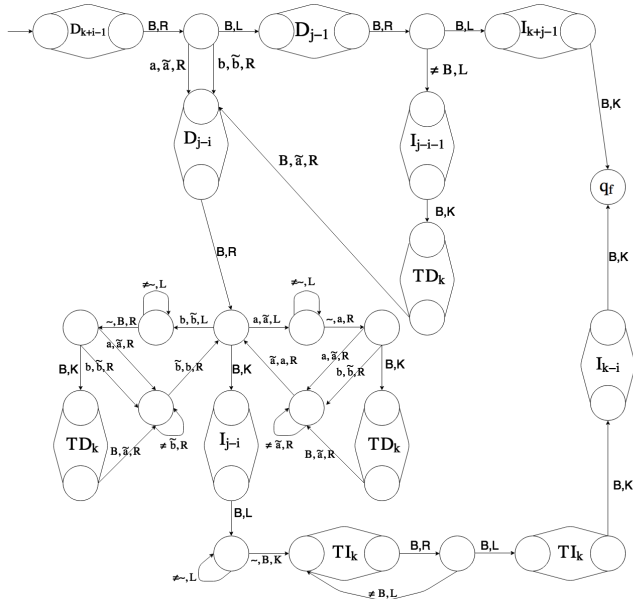
$M_{i,k}^a$ puede ser la siguiente máquina:



Una posible forma de diseñar la máquina $IF_{i,k}^a$ es la siguiente:



Una posible forma de diseñar la máquina $M_{i \leftarrow j}^{*,k}$ para el caso $\Sigma = \{a, b\}$ y $i < j$, es la siguiente:



Probaremos

El paradigma computacional de Turing es por lo menos tan expresivo como el paradigma imperativo dado por el lenguaje S^Σ , es decir, probaremos que toda función Σ -computable es Σ -Turing computable.

Antes un lema:

Lemma

*Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \Sigma^*$ es Σ -computable, entonces existe un programa Q , el cual computa a f y cumple con las siguientes propiedades:*

- 1) En Q no hay instrucciones de la forma GOTO $L\bar{i}$, ni de la forma $L\bar{j}$ GOTO $L\bar{i}$.*
- 2) Cuando Q termina partiendo de un estado cualquiera dado, el estado alcanzado es tal que las variables numéricas tienen todas el valor 0 y las alfabéticas tienen, todas excepto $P1$, el valor ε .*

Proof.

Sean:

- \mathcal{P} un programa que compute a f
- $r \in \mathbb{N}$ tal que $r \geq N(\mathcal{P}), n, m$
- $\tilde{\mathcal{P}}$ el resultado de reemplazar en \mathcal{P} cada instrucción de la forma $\alpha \text{GOTO } L\bar{i}$ con $\alpha \in \{\varepsilon\} \cup \{L\bar{j} : j \in \mathbb{N}\}$ por $\alpha \text{IF } N\bar{r} \neq 0 \text{ GOTO } L\bar{i}$.

Ahora, sea \mathcal{Q} el siguiente programa:

$$N\bar{r} \leftarrow N\bar{r} + 1$$

$$\tilde{\mathcal{P}}$$

$$N1 \leftarrow 0$$

$$\vdots$$

$$N\bar{r} \leftarrow 0$$

$$P2 \leftarrow \varepsilon$$

$$\vdots$$

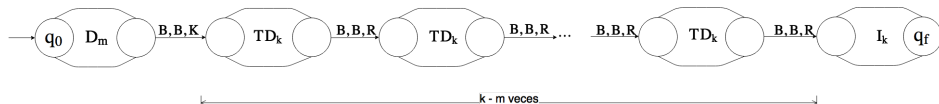
$$P\bar{r} \leftarrow \varepsilon$$

Theorem

Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ es Σ -computable, entonces f es Σ -Turing computable.

Proof

Supongamos $O = \Sigma^*$. Por el Lema anterior, existe $\mathcal{P} \in \text{Pro}^\Sigma$ el cual computa f y tiene las propiedades (1) y (2). Sea $k = \max\{n, m, N(\mathcal{P})\}$ y sea M_{sim} la máquina de Turing con unit que simula a \mathcal{P} respecto de k . Como puede observarse, la máquina M_{sim} , no necesariamente computará a f . Sea M_1 la siguiente máquina:



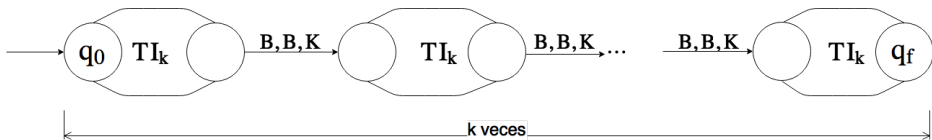
Observación

Cuando $n = 0$ debemos interpretar que $D_0 = (\{q_0, q_f\}, \Gamma, \Sigma, \delta, q_0, B, \vdash, \{q_f\})$, con $\delta(q_0, B) = \{(q_f, B, K)\}$ y $\delta = \emptyset$ en cualquier otro caso.

Nótese que M_1 cumple que, para cada $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m}$

$$\lfloor q_0 B \vdash^{x_1} B \dots B \vdash^{x_n} B \alpha_1 B \dots B \alpha_m B \rfloor \vdash^* \lfloor q_f B \vdash^{x_1} B \dots B \vdash^{x_n} B^{k-n} B \alpha_1 B \dots B \alpha_m B \rfloor$$

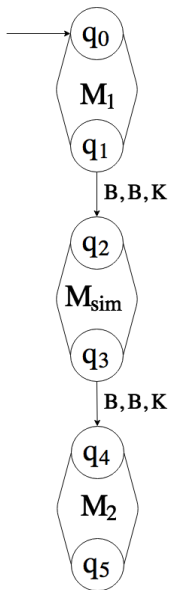
Sea M_2 la siguiente máquina:



Nótese que M_2 cumple que para cada $\alpha \in \Sigma^*$

$$\left[q_0 B^{k+1} \alpha \right] \vdash^* \left[q_f B \alpha \right]$$

Sea M la máquina dada por el siguiente diagrama:



Supongamos que $(\vec{x}, \vec{\alpha}) \in (\omega^n \times \Sigma^{*m}) - D_f$. Debemos ver que M no termina partiendo de:

$$[q_0 B \mid^{x_1} B \dots B \mid^{x_n} B \alpha_1 B \dots B \alpha_m B] \quad (*)$$

Primero notemos que, ya que \mathcal{P} computa a f , tenemos que \mathcal{P} no termina partiendo de $\|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\|$ por lo cual \mathcal{P} no termina partiendo de

$$\|x_1, \dots, x_n, \underbrace{0, \dots, 0}_{k-n}, \alpha_1, \dots, \alpha_m, \underbrace{\varepsilon, \dots, \varepsilon}_{k-m}\|$$

lo cual implica, por el Lema 1, que M_{sim} no termina partiendo de:

$$[q_0 B \mid^{x_1} B \dots B \mid^{x_n} B^{k-n} B \alpha_1 B \dots B \alpha_m B] \quad (**)$$

Ahora, nótese que si hacemos funcionar a M desde la descripción instantánea dada en (*), llegaremos indefectiblemente a la siguiente descripción instantánea:

$$\left[q_2 B^{x_1} B \dots B^{x_n} B^{k-n} B_{\alpha_1} B \dots B_{\alpha_m} B \right]$$

entonces (**) nos dice que al seguir trabajando M , la máquina M nunca terminará.

Para terminar de ver que M computa a f , tomemos $(\vec{x}, \vec{\alpha}) \in D_f$ y veamos que

$$\left[q_0 B^{x_1} B \dots B^{x_n} B_{\alpha_1} B \dots B_{\alpha_m} B \right] \stackrel{*}{\vdash}_M \left[q_5 Bf(\vec{x}, \vec{\alpha}) \right]$$

y que la máquina M se detiene en $\left[q_5 Bf(\vec{x}, \vec{\alpha}) \right]$.

- La máquina M se detiene en $\lfloor q_5 B f(\vec{x}, \vec{\alpha}) \rfloor$, ya que q_5 es el estado final de una copia de M_2 y por lo tanto no sale ninguna flecha desde él.
- Ya que \mathcal{P} computa a f y tiene la propiedad (2) del Lema 2, tenemos que \mathcal{P} termina partiendo de $\|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\|$ y llega al estado $\|f(\vec{x}, \vec{\alpha})\|$, o lo que es lo mismo, \mathcal{P} termina partiendo de

$$\|x_1, \dots, x_n, \overbrace{0, \dots, 0}^{k-n}, \alpha_1, \dots, \alpha_m, \overbrace{\varepsilon, \dots, \varepsilon}^{k-m}\|$$

y llega al estado

$$\|\overbrace{0, \dots, 0}^k, f(\vec{x}, \vec{\alpha}), \overbrace{\varepsilon, \dots, \varepsilon}^{k-1}\|$$

Pero entonces el Lema 1 nos dice que:

$$\lfloor q_0 B \mid^{x_1} B \dots B \mid^{x_n} B^{k-n} B \alpha_1 B \dots B \alpha_m B \rfloor \vdash_{M_{sim}}^* \lfloor q_f B^{k+1} f(\vec{x}, \vec{\alpha}) \rfloor \quad (***)$$

Como ya lo vimos, si hacemos funcionar a M desde:

$$\lfloor q_0 B^{x_1} B \dots B^{x_n} B^{\alpha_1} B \dots B^{\alpha_m} B \rfloor$$

llegaremos, vía la copia de M_1 dentro de M , indefectiblemente a la siguiente descripción instantánea

$$\lfloor q_2 B^{x_1} B \dots B^{x_n} B^{k-n} B^{\alpha_1} B \dots B^{\alpha_m} B \rfloor$$

Luego, (***) nos dice que, vía la copia de M_{sim} dentro de M , llegaremos a $\lfloor q_3 B^{k+1} f(\vec{x}, \vec{\alpha}) \rfloor$ e inmediatamente a $\lfloor q_4 B^{k+1} f(\vec{x}, \vec{\alpha}) \rfloor$.

Finalmente, vía la copia de M_2 dentro de M , llegaremos a $\lfloor q_5 B f(\vec{x}, \vec{\alpha}) \rfloor$, lo cual termina de demostrar que M computa a f .