Turing vence a von Neumann

Agustín Curto

FaMAF

2017

Notación: dados $x_1, \ldots, x_n \in \omega$ y $\alpha_1, \ldots, \alpha_m \in \Sigma^*$, con $n, m \in \omega$, usaremos:

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

para denotar el estado

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

Notese que por ejemplo:

$$\begin{aligned} \|x\| &= ((x, 0, \ldots), (\varepsilon, \ldots)) \text{ Para } n = 1, m = 0 \\ \|\Diamond\| &= ((0, \ldots), (\varepsilon, \ldots)) \text{ Para } n = m = 0 \end{aligned}$$

Ademas es claro que:

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

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

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

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

$$N(\mathcal{P}) = \text{menor } k \in \mathbf{N} \text{ tal que las variables que ocurren en } \mathcal{P} \text{ estan todas en la lista } \mathrm{N1}, \ldots, \mathrm{N}\bar{k}, \mathrm{P1}, \ldots, \mathrm{P}\bar{k}$$

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

$$\begin{array}{ll} \mathrm{L1} & \mathrm{N4} \leftarrow \mathrm{N4} + 1 \\ & \mathrm{P1} \leftarrow \mathrm{P1}.\& \\ & \mathrm{IF\ N1} \neq 0\ \mathrm{GOTO\ L1} \end{array}$$

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

Sea $\mathcal{P}$ un programa y sea k fijo y mayor o igual a $N(\mathcal{P})$. A continuación describiremos como puede construirse una maquina de Turing la cual simulara a $\mathcal{P}$. La construccion de la maquina simuladora dependera de $\mathcal{P}$ y de k. Notese que cuando $\mathcal{P}$ se corre desde algun estado de la forma

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

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

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

es decir en todos ellos las variables con indice mayor que k valen 0 o ε. La razon es simple: ya que en $\mathcal{P}$ no figuran las variables

$$\mathrm{N}\overline{k+1}, \mathrm{N}\overline{k+2}, ...$$
$$\mathrm{P}\overline{k+1}, \mathrm{P}\overline{k+2}, ...$$

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

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

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

lo representaremos en la cinta de la siguiente manera

$$B\ |^{x_1}\ ...B\ |^{x_k}\ B\alpha_1...B\alpha_k BBBB....$$

Por ejemplo consideremos el programa $\mathcal{P}$ mostrado recien y fijemos $k = 6$. Entonces al estado

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

lo representaremos en la cinta de la siguiente manera

$$B\ |||\ B\ ||\ B|||||BB\ ||||\ B\ ||\ B\&B\&\&BB\#\&B\#B\#\#\#BBBBB....$$

A lo que queda entre dos blancos consecutivos (es decir que no hay ningun blanco entre ellos) lo llamaremos "bloque", por ejemplo en la cinta de arriba tenemos que los primeros 12 bloques son

$$\text{III} \quad \text{II} \quad \text{IIIII} \quad \varepsilon \quad \text{IIII} \quad \text{II} \quad \& \quad \&\& \quad \varepsilon \quad \#\& \quad \# \quad \#\#\#$$

y despues los bloques siguientes (que son infinitos ya que la cinta es infinita hacia la derecha) son todos iguales a ε.
Una observacion importante es que esta forma de representacion de estados en la cinta depende del k elejido, es decir si tomaramos otro k, por ejemplo $k = 9$, entonces el estado anterior se representaria de otra forma en la cinta.

Armaremos la máquina simuladora como concatenación de máquinas, cada una de las cuales simularán, vía la representación anterior, el funcionamiento de las distintas instrucciones de $\mathcal{P}$. Para esto, a continuacion describiremos para los distintos tipos de instrucciones posibles de $\mathcal{P}$, sus respectivas maquinas asociadas. Asumiremos que en $\mathcal{P}$ no hay instrucciones de la forma $\mathrm{GOTO\ L}\bar{m}$ ni de la forma $\mathrm{L}\bar{n}\ \mathrm{GOTO\ L}\bar{m}$. Esto simplificara un poco la costruccion de la maquina simuladora y de hecho lo podemos hacer ya que toda funcion Σ-computable puede ser computada por un programa sin este tipo de instrucciones, tal como lo veremos en un lema mas adelante.
En esta etapa solo describiremos que propiedades tendra que tener cada maquina simuladora de cada tipo posible de instruccion, y mas adelante mostraremos como pueden ser construidas efectivamente dichas maquinas.
Todas las maquinas descriptas tendran a $\shortmid$ como unit y a B como blanco, tendran a Σ como su alfabeto terminal y su alfabeto mayor sera $\Gamma = \Sigma \cup \{B, \shortmid\} \cup \{\tilde{a} : a \in \Sigma \cup \{\shortmid\}\}$. Ademas tendran uno o dos estados finales con la propiedad de que si q es un estado final, entonces $\delta(q, \sigma) = \emptyset$, para cada $\sigma \in \Gamma$. Esta propiedad es importante ya que nos permitira concatenar pares de dichas maquinas identificando algun estado final de la primera con el inicial de la segunda.

- Para $1 \leq i \leq k$, sea M_i^+ una máquina tal que:

$$\begin{array}{lcl} B\,|^{x_1}\,B\ldots B\,|^{x_k}\,B\alpha_1 B\ldots B\alpha_k & \overset{*}{\vdash} & B\,|^{x_1}\,B\ldots B\,|^{x_{i-1}}\,B\,|^{x_i+1}\,B\,|^{x_{i+1}}\,\ldots \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

- Para $1 \leq i \leq k$, sea $M_i^{\dot{-}}$ una máquina tal que:

$$\begin{array}{lcl} B\,|^{x_1}\,B\ldots B\,|^{x_k}\,B\alpha_1 B\ldots B\alpha_k & \overset{*}{\vdash} & B\,|^{x_1}\,B\ldots B\,|^{x_{i-1}}\,B\,|^{x_i\dot{-}1}\,B\,|^{x_{i+1}}\,\ldots \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

- Para $1 \leq i \leq k$ y $a \in \Sigma$, sea M_i^a una máquina tal que:

$$\begin{array}{lcl} B\,|^{x_1}\,B\ldots B\,|^{x_k}\,B\alpha_1 B\ldots B\alpha_k & \overset{*}{\vdash} & B\,|^{x_1}\,B\ldots B\,|^{x_k}\,B\alpha_1 B\ldots B\alpha_{i-1} B\alpha \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

- Para $1 \leq i \leq k$, sea $M_i^{\curvearrowright}$ una máquina tal que:

$$\begin{array}{l} B \shortmid^{x_1} B \ldots B \shortmid^{x_k} B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B \shortmid^{x_1} B \ldots B \shortmid^{x_k} B\alpha_1 B \ldots B\alpha_{i-1} B^{\curvearrowright} \\ \uparrow \\ q_f \end{array}$$

- Para $j = 1, \ldots, k$, y $a \in \Sigma$, sea IF_j^a una máquina con dos estados finales q_{si} y q_{no} tal que:
  Si α_j comienza con a, entonces:

$$\begin{array}{l} B \shortmid^{x_1} B \ldots B \shortmid^{x_k} B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B \shortmid^{x_1} B \ldots B \shortmid^{x_k} B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_{si} \end{array}$$

- Caso contrario:

$$\begin{array}{l} B \shortmid^{x_1} B \ldots B \shortmid^{x_k} B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B \shortmid^{x_1} B \ldots B \shortmid^{x_k} B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_{no} \end{array}$$

Análogamente, para $j = 1, \ldots, k$, sea IF_j una máquina tal que:

- Si $x_j \neq 0$, entonces:

$$\begin{array}{l} B\ |^{x_1}\ B \ldots B\ |^{x_k}\ B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \quad \overset{*}{\vdash} \quad \begin{array}{l} B\ |^{x_1}\ B \ldots B\ |^{x_k}\ B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_{si} \end{array}$$

- Si $x_j = 0$, entonces:

$$\begin{array}{l} B\ |^{x_1}\ B \ldots B\ |^{x_k}\ B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \quad \overset{*}{\vdash} \quad \begin{array}{l} B\ |^{x_1}\ B \ldots B\ |^{x_k}\ B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_{no} \end{array}$$

- Para $1 \leq i, j \leq k$, sea $M^*_{i \leftarrow j}$ una máquina tal que:

$$\begin{array}{l} B\ |^{x_1}\ B \ldots B\ |^{x_k}\ B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \quad \overset{*}{\vdash} \quad \begin{array}{l} B\ |^{x_1}\ B \ldots B\ |^{x_k}\ B\alpha_1 B \ldots B\alpha_{i-1} B\alpha_j \\ \uparrow \\ q_f \end{array}$$

- Para $1 \leq i, j \leq k$, sea $M^{\#}_{i \leftarrow j}$ una máquina tal que:

$$\begin{array}{l} B \,|^{x_1}\, B \ldots B \,|^{x_k}\, B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \quad \overset{*}{\vdash} \quad \begin{array}{l} B \,|^{x_1}\, B \ldots B \,|^{x_{i-1}}\, B \,|^{x_j}\, B \,|^{x_{i+1}}\, B \ldots \\ \uparrow \\ q_f \end{array}$$

- Para $1 \leq i \leq k$, sea $M_{i \leftarrow 0}$ una máquina tal que:

$$\begin{array}{l} B \,|^{x_1}\, B \ldots B \,|^{x_k}\, B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \quad \overset{*}{\vdash} \quad \begin{array}{l} B \,|^{x_1}\, B \ldots B \,|^{x_{i-1}}\, BB \,|^{x_{i+1}}\, B \ldots B \,|^{x_k} \\ \uparrow \\ q_f \end{array}$$

- Para $1 \leq i \leq k$, sea $M_{i \leftarrow \varepsilon}$ una máquina tal que:

$$\begin{array}{l} B \,|^{x_1}\, B \ldots B \,|^{x_k}\, B\alpha_1 B \ldots B\alpha_k \\ \uparrow \\ q_0 \end{array} \quad \overset{*}{\vdash} \quad \begin{array}{l} B \,|^{x_1}\, B \ldots B \,|^{x_k}\, B\alpha_1 B \ldots B\alpha_{i-1} BB \\ \uparrow \\ q_f \end{array}$$

A continuacion veremos un ejemplo de como se arma la maquina simuladora de un programa dado. Sea $\Sigma = \{\&, \#\}$ y sea $\mathcal{P}$ el siguiente programa

$$
\begin{array}{ll}
\mathrm{L3} & \mathrm{N4} \leftarrow \mathrm{N4} + 1 \\
& \mathrm{P1} \leftarrow \ ^{\frown}\mathrm{P1} \\
& \mathrm{IF\ P1\ BEGINS\ \&\ GOTO\ L3} \\
& \mathrm{P3} \leftarrow \mathrm{P3}.\#
\end{array}
$$

Tomemos $k = 5$. Es claro que $k \geq N(\mathcal{P}) = 4$. A la maquina que simulara a $\mathcal{P}$ respecto de k, la llamaremos M_{sim} y sera la siguiente:

Figura 8

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 sucesion de descripciones instantaneas:

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

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

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

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

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

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

$$q_0 B \quad | \quad {}^{2}B \, | \, BB \, |^{5} \, B \, |^{3} \, B\#\&\#\#BB\&\&B\#\&B\#B$$

$$q_1 B \quad | \quad {}^{2}B \, | \, BB \, |^{6} \, B \, |^{3} \, B\#\&\#\#BB\&\&B\#\&B\#B$$
$$q_2 B \quad | \quad {}^{2}B \, | \, BB \, |^{6} \, B \, |^{3} \, B\#\&\#\#BB\&\&B\#\&B\#B$$

$$q_3 B \quad | \quad {}^{2}B \, | \, BB \, |^{6} \, B \, |^{3} \, B\&\#\#BB\&\&B\#\&B\#B$$
$$q_4 B \quad | \quad {}^{2}B \, | \, BB \, |^{6} \, B \, |^{3} \, B\&\#\#BB\&\&B\#\&B\#B$$

$$q_{si} B \quad | \quad {}^{2}B \, | \, BB \, |^{6} \, B \, |^{3} \, B\&\#\#BB\&\&B\#\&B\#B$$
$$q_0 B \quad | \quad {}^{2}B \, | \, BB \, |^{6} \, B \, |^{3} \, B\&\#\#BB\&\&B\#\&B\#B$$

$$q_1 B \quad | \quad {}^{2}B \, | \, BB \, |^{7} \, B \, |^{3} \, B\&\#\#BB\&\&B\#\&B\#B$$
$$q_2 B \quad | \quad {}^{2}B \, | \, BB \, |^{7} \, B \, |^{3} \, B\&\#\#BB\&\&B\#\&B\#B$$

$$q_3 B \quad | \quad {}^{2}B \, | \, BB \, |^{7} \, B \, |^{3} \, B\#\#BB\&\&B\#\&B\#B$$

Ahora describiremos en general como puede armarse la maquina simuladora de $\mathcal{P}$, respecto de k. Supongamos que $\mathcal{P} = I_1 ... I_n$. Para cada $i = 1, ..., n$, llamaremos M_i a la maquina que simulara el efecto que produce la instruccion I_i, es decir tomemos

Proof.

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

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

Figura 5

entendiendo que en el caso en que M_i tiene un solo estado final, este esta representado por el circulo de abajo a la izquierda y en el caso en que M_i tiene dos estados finales, el estado final representado con lineas punteadas (es decir, el de la derecha) corresponde al estado q_{si} y el otro al estado q_{no}.

Para armar la maquina que simulara a $\mathcal{P}$ hacemos lo siguiente. Primero unimos las maquinas $M_1, ..., M_n$ de la siguiente manera

Nueva Figura 6

Luego para cada i tal que $Bas(I_i)$ es de la forma $\alpha \mathrm{GOTO\ L}\bar{m}$, 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 instruccion que tiene label $\mathrm{L}\bar{m}$.

Lemma

Sea $\mathcal{P} \in \mathrm{Pro}^{\Sigma}$ *y sea* $k \geq N(\mathcal{P})$. *Supongamos que en* $\mathcal{P}$ *no hay instrucciones de la forma* $\mathrm{GOTO\ L}\bar{m}$ *ni de la forma* $\mathrm{L}\bar{n}\ \mathrm{GOTO\ L}\bar{m}$. *Para cada* $a \in \Sigma \cup \{|\}$, *sea* $\tilde{a}$ *un nuevo simbolo. Sea* $\Gamma = \Sigma \cup \{B, |\} \cup \{\tilde{a} : a \in \Sigma \cup \{|\}\}$. *Entonces hay una maquina de Turing deterministica 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* $x_1, ..., x_k \in \omega$ *y* $\alpha_1, ..., \alpha_k \in \Sigma^*$, *el programa* $\mathcal{P}$ *se detiene partiendo del estado*

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

sii M *se detiene partiendo de la descripcion instantanea*

$$\lfloor q_0 B\ |^{x_1}\ B ... B\ |^{x_k}\ B\alpha_1 B ... B\alpha_k B \rfloor$$

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

En esta etapa nos dedicaremos a explicar como construir las distintas maquinas simuladoras de instrucciones.
Para poder hacer concretamente las maquinas recien descriptas deberemos diseñar antes algunas maquinas auxiliares. Para cada $j \geq 1$, sea D_j la maquina siguiente

Figura 1

Notese que

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

siempre que $\alpha, \gamma \in \Gamma^*$, $\beta_1, ..., \beta_j \in (\Gamma - \{B\})^*$. Es decir la maquina D_j lo unico que hace es mover el cabezal desde el blanco de la izquierda de un bloque determinado, exactamente j bloques a la derecha
Analogamente I_j sera una maquina que desplaza el cabezal j bloques a la izquierda del blanco que esta escaneando. Es decir I_j cumplira que

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

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

Para $j \geq 1$, sea TD_j una maquina 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. Es decir la maquina TD_j corre un espacio a la derecha todo el segmento γ y agrega un blanco en el espacio que se genera a la izquierda. Por ejemplo, para el caso de $\Sigma = \{\&, \#\}$ podemos tomar TD_3 igual a la siguiente maquina

Nueva Figura 3

Analogamente, para $j \geq 1$, sea TI_j una maquina 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. Es decir la maquina TI_j corre un espacio a la izquierda todo el segmaneto γ (por lo cual en el lugar de σ queda el primer simbolo de γ). Dejamos al lector la construccion de por ejemplo TI_3 para $\Sigma = \{\&, \#\}$
Teniendo las maquinas auxiliares antes definidas podemos combinarlas para obtener las maquinas simuladoras de instrucciones. Por ejemplo $M^a_{i,k}$ puede ser la siguiente maquina

Figura 4

Una posible forma de diseñar la maquina $IF^a_{i,k}$ es la siguiente

Figura 2

Una posible forma de diseñar la maquina $M^{*,k}_{i\leftarrow j}$ para el caso $\Sigma = \{\&, \#\}$ y $i < j$, es la siguiente

Nueva Figura 7

En lo que sigue probaremos que el paradigma computacional de Turing es por lo menos tan expresivo como el paradigma imperativo dado por el lenguaje $\mathcal{S}^{\Sigma}$, es decir probaremos que toda funcion Σ-computable es Σ-Turing computable. Antes un lema

Lemma

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

(1) *En $\mathcal{Q}$ no hay instrucciones de la forma $\mathrm{GOTO\ L}\bar{i}$ ni de la forma $\mathrm{L}\bar{j}\ \mathrm{GOTO\ L}\bar{i}$*

(2) *Cuando $\mathcal{Q}$ termina partiendo de un estado cualquiera dado, el estado alcansado es tal que las variables numericas tienen todas el valor 0 y las alfabeticas tienen todas exepto $\mathrm{P1}$ el valor ε.*

Proof.

Sea $\mathcal{P}$ un programa que compute a f. Sea $r \in \mathbf{N}$ tal que $r \geq N(\mathcal{P}), n, m$. Sea $\tilde{\mathcal{P}}$ el resultado de reemplazar en $\mathcal{P}$ cada instruccion de la forma

$$\alpha \mathrm{GOTO\ L}\bar{i}$$

con $\alpha \in \{\varepsilon\} \cup \{\mathrm{L}\bar{j} : j \in \mathbf{N}\}$ por $\alpha \mathrm{IF\ N}\bar{r} \neq 0\ \mathrm{GOTO\ L}\bar{i}$. Ahora sea $\mathcal{Q}$ el siguiente programa

$$\begin{aligned}
&\mathrm{N}\bar{r} \leftarrow \mathrm{N}\bar{r} + 1\\
&\tilde{\mathcal{P}}\\
&\mathrm{N1} \leftarrow 0\\
&\vdots\\
&\mathrm{N}\bar{r} \leftarrow 0\\
&\mathrm{P2} \leftarrow \varepsilon\\
&\vdots\\
&\mathrm{P}\bar{r} \leftarrow \varepsilon
\end{aligned}$$

Es facil ver que $\mathcal{Q}$ tiene las propiedades (1) y (2). □

Por supuesto, hay un lema analogo para el caso en que f llega a ω en lugar de llegar a Σ^*

Ahora si, el anunciado del teorema:

Theorem

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

Proof.

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

Figura 9

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

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

$$\lfloor q_0 B \shortmid^{x_1} B \ldots B \shortmid^{x_n} B\alpha_1 B \ldots B\alpha_m B \rfloor \overset{*}{\vdash} \left\lfloor q_f B \shortmid^{x_1} B \ldots B \shortmid^{x_n} B^{k-n} B\alpha_1 B \ldots B\alpha_m B \right\rfloor$$

Notese que en la confeccion de M_1, para el caso $m > 0$ podriamos haber usado directamente la TD_m en lugar de usar TD_m.

Sea M_2 la siguiente maquina

Figura 10