TOMA 0: A continuacion introduciremos una notacion que aun no esta en el apunte y nos sera util. Dados $x_1, ..., x_n \in \omega$ y $\alpha_1, ..., \alpha_m \in \Sigma^*$, con $n, m \in \omega$, usaremos

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

para denotar el estado

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

Notese que por ejemplo

$$\|x\| = ((x, 0, ...), (\varepsilon, ...))$$

(es el caso $n = 1$ y $m = 0$) y tambien

$$\|\Diamond\| = ((0, ...), (\varepsilon, ...))$$

(es el caso $n = m = 0$). Ademas es claro que

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

cualesquiera sean $i, j \in \omega$

esta toma debe estar por separado asi la eliminamos cuando se agregue la def al apunte

TOMA 1: En este video probaremos que toda funcion Σ-computable es Σ-Turing computable, es decir probaremos que si f es una funcion Σ-mixta que es computada por un programa $\mathcal{P} \in \text{Pro}^{\Sigma}$, entonces hay una maquina de Turing deterministica con unit M la cual computa a f. Para esto probaremos un resultado general que nos enseñara a simular el comportamiento de un programa con una maquina de Turing. Es importante notar que la simulacion que nos interesa que haga la maquina simuladora no es a nivel de la funcion que computa el programa sino a un nivel mas general, es decir nos interesa que simule a dicho programa como transformador de estados.

Necesitaremos una definicion. Dado $\mathcal{P} \in \text{Pro}^{\Sigma}$, definamos

$$\begin{array}{rcl} N(\mathcal{P}) & = & \text{menor } k \in \mathbf{N} \text{ tal que las variables que ocurren en } \mathcal{P} \\ & & \text{estan todas en la lista } \text{N1}, ..., \text{N}\bar{k}, \text{P1}, ..., \text{P}\bar{k} \end{array}$$

Por ejemplo si $\mathcal{P}$ es el siguiente programa (aqui $\Sigma = \{\&, \#\}$)

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

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

Sea $\mathcal{P}$ un programa y sea k fijo y mayor o igual a $N(\mathcal{P})$. A continuacion 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.

La maquina simuladora que construiremos simulara a $\mathcal{P}$ en cuanto a su funcionamiento cuando partimos de estados de la anterior forma. 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

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

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

$$|||\quad ||\quad |||||\quad \varepsilon\quad ||||\quad ||\quad \&\quad \&\&\quad \varepsilon\quad \#\&\quad \#\quad \#\#\#$$

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.

TOMA 2: Armaremos la maquina simuladora como concatenacion de maquinas cada una de las cuales simularan, via la representacion 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 GOTO L$\bar{m}$ ni de la forma L$\bar{n}$ 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 $M^+_{i,k}$ una maquina tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

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

donde q_0 es el estado inicial y q_f es el unico estado final de $M^+_{i,k}$. Es claro que la maquina $M^+_{i,k}$ simula la instruccion N$\bar{\imath} \leftarrow$ N$\bar{\imath} + 1$, via la representacion de estados en la cinta con respecto a k.

Para $1 \leq i \leq k$, sea $M^{\dot{-}}_{i,k}$ una maquina tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

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

donde q_0 es el estado inicial y q_f es el unico estado final de $M^{\dot{-}}_{i,k}$. Es claro que la maquina $M^{\dot{-}}_{i,k}$ simula la instruccion P$\bar{\imath} \leftarrow$ P$\bar{\imath} \dot{-} 1$, via la representacion de estados en la cinta con respecto a k.

Para $1 \leq i \leq k$ y $a \in \Sigma$, sea $M^a_{i,k}$ una maquina tal que cualesquiera sean

$x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

$$\begin{array}{lcl} B\,{\shortmid}^{x_1}\,...B\,{\shortmid}^{x_k}\,B\alpha_1...B\alpha_k & \overset{*}{\vdash} & B\,{\shortmid}^{x_1}\,...B\,{\shortmid}^{x_k}\,B\alpha_1...B\alpha_{i-1}B\alpha_i a B\alpha_{i+1}...B\alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

donde q_0 es el estado inicial y q_f es el unico estado final de $M^a_{i,k}$. Es claro que la maquina $M^a_{i,k}$ simula la instruccion $\mathrm{P}\bar{\imath} \leftarrow \mathrm{P}\bar{\imath}.a$, via la representacion de estados en la cinta con respecto a k.

Para $1 \leq i \leq k$, sea $M^{\curvearrowright}_{i,k}$ una maquina tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

$$\begin{array}{lcl} B\,{\shortmid}^{x_1}\,...B\,{\shortmid}^{x_k}\,B\alpha_1...B\alpha_k & \overset{*}{\vdash} & B\,{\shortmid}^{x_1}\,...B\,{\shortmid}^{x_k}\,B\alpha_1...B\alpha_{i-1}B{\curvearrowright}\alpha_i B\alpha_{i+1}...B\alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

donde q_0 es el estado inicial y q_f es el unico estado final de $M^{\curvearrowright}_{i,k}$. Es claro que la maquina $M^{\curvearrowright}_{i,k}$ simula la instruccion $\mathrm{P}\bar{\imath} \leftarrow {\curvearrowright}\mathrm{P}\bar{\imath}$, via la representacion de estados en la cinta con respecto a k.

Para $1 \leq i, j \leq k$, sea $M^{\#,k}_{i \leftarrow j}$ una maquina tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

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

donde q_0 es el estado inicial y q_f es el unico estado final de $M^{\#,k}_{i \leftarrow j}$. Es claro que la maquina $M^{\#,k}_{i \leftarrow j}$ simula la instruccion $\mathrm{N}\bar{\imath} \leftarrow \mathrm{N}\bar{\jmath}$, via la representacion de estados en la cinta con respecto a k.

Para $1 \leq i, j \leq k$, sea $M^{*,k}_{i \leftarrow j}$ una maquina tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

$$\begin{array}{lcl} B\,{\shortmid}^{x_1}\,...B\,{\shortmid}^{x_k}\,B\alpha_1...B\alpha_k & \overset{*}{\vdash} & B\,{\shortmid}^{x_1}\,...B\,{\shortmid}^{x_k}\,B\alpha_1...B\alpha_{i-1}B\alpha_j B\alpha_{i+1}...B\alpha_k \\ \uparrow & & \uparrow \\ q_0 & & q_f \end{array}$$

donde q_0 es el estado inicial y q_f es el unico estado final de $M^{*,k}_{i \leftarrow j}$. Es claro que la maquina $M^{*,k}_{i \leftarrow j}$ simula la instruccion $\mathrm{P}\bar{\imath} \leftarrow \mathrm{P}\bar{\jmath}$, via la representacion de estados en la cinta con respecto a k.

Para $1 \leq i \leq k$, sea $M^k_{i \leftarrow 0}$ una maquina tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

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

donde q_0 es el estado inicial y q_f es el unico estado final de $M^k_{i\leftarrow 0}$. Es claro que la maquina $M^k_{i\leftarrow 0}$ simula la instruccion N$\bar{\imath} \leftarrow 0$, via la representacion de estados en la cinta con respecto a k.

Para $1 \leq i \leq k$, sea $M^k_{i\leftarrow \varepsilon}$ una maquina tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$:

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

donde q_0 es el estado inicial y q_f es el unico estado final de $M^k_{i\leftarrow \varepsilon}$. Es claro que la maquina $M^k_{i\leftarrow \varepsilon}$ simula la instruccion P$\bar{\imath} \leftarrow \varepsilon$, via la representacion de estados en la cinta con respecto a k.

Sea

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

con $\delta(q_0, B) = \{(q_f, B, K)\}$ y $\delta = \emptyset$ en cualquier otro caso. Es claro que la maquina M_{SKIP} simula la instruccion SKIP, via la representacion de estados en la cinta con respecto a k (cualquiera sea el k).

Para $1 \leq j \leq k$, sea $IF_{j,k}$ una maquina con estado inicial q_0 y dos estados finales q_{si} y q_{no} tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$, si $x_j \neq 0$, entonces

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

y si $x_j = 0$, entonces

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

Para $1 \leq i \leq k$ y $a \in \Sigma$, sea $IF^a_{j,k}$ una maquina con estado inicial q_0 y dos estados finales q_{si} y q_{no} tal que cualesquiera sean $x_1, ..., x_k \in \omega$ y $\alpha_1, ..., \alpha_k \in \Sigma^*$, si α_j comienza con a, entonces

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

y en caso contrario

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

TOMA 3: 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

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

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:

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

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

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

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

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

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

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

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

Si hacemos funcionar a M_{sim} desde $q_0 B \vdash^2 B \vdash BB \vdash^5 B \vdash^3 B\#\&\#\#BB\&\&B\#\&B\#B$ obtendremos una sucesion de descripciones instantaneas dentro de la cual estara

la siguiente subsucesion que se corresponde con las descripciones instantaneas de la computacion anterior.

$$
\begin{array}{rcl}
q_0B & | & {}^2B \shortmid BB \shortmid^5 B \shortmid^3 B\#\&\#\#BB\&\&B\#\&B\#B \\
\\
q_1B & | & {}^2B \shortmid BB \shortmid^6 B \shortmid^3 B\#\&\#\#BB\&\&B\#\&B\#B \\
q_2B & | & {}^2B \shortmid BB \shortmid^6 B \shortmid^3 B\#\&\#\#BB\&\&B\#\&B\#B \\
\\
q_3B & | & {}^2B \shortmid BB \shortmid^6 B \shortmid^3 B\&\#\#BB\&\&B\#\&B\#B \\
q_4B & | & {}^2B \shortmid BB \shortmid^6 B \shortmid^3 B\&\#\#BB\&\&B\#\&B\#B \\
\\
q_{si}B & | & {}^2B \shortmid BB \shortmid^6 B \shortmid^3 B\&\#\#BB\&\&B\#\&B\#B \\
q_0B & | & {}^2B \shortmid BB \shortmid^6 B \shortmid^3 B\&\#\#BB\&\&B\#\&B\#B \\
\\
q_1B & | & {}^2B \shortmid BB \shortmid^7 B \shortmid^3 B\&\#\#BB\&\&B\#\&B\#B \\
q_2B & | & {}^2B \shortmid BB \shortmid^7 B \shortmid^3 B\&\#\#BB\&\&B\#\&B\#B \\
\\
q_3B & | & {}^2B \shortmid BB \shortmid^7 B \shortmid^3 B\#\#BB\&\&B\#\&B\#B \\
q_4B & | & {}^2B \shortmid BB \shortmid^7 B \shortmid^3 B\#\#BB\&\&B\#\&B\#B \\
\\
q_{no}B & | & {}^2B \shortmid BB \shortmid^7 B \shortmid^3 B\#\#BB\&\&B\#\&B\#B \\
q_5B & | & {}^2B \shortmid BB \shortmid^7 B \shortmid^3 B\#\#BB\&\&B\#\&B\#B \\
\\
q_6B & | & {}^2B \shortmid BB \shortmid^7 B \shortmid^3 B\#\#BB\&\&\#B\#\&B\#B
\end{array}
$$

TOMA 4: 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 αGOTO 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 L$\bar{m}$.

TOMA 5: A continuacion enunciaremos en forma de lema la existencia de la maquina simuladora y de las propiedades esenciales que usaremos luego para probar que toda funcion Σ-computable es Σ-Turing computable

Lemma 1 *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* GOTO L$\bar{m}$ *ni de la forma* L$\bar{n}$ 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 determinista 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*

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

y llega al estado

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

entonces

$$\lfloor q_0 B\ |^{x_1}\ B...B\ |^{x_k}\ B\alpha_1 B...B\alpha_k B \rfloor \overset{*}{\underset{M}{\vdash}} \lfloor q_f B\ |^{y_1}\ B...B\ |^{y_k}\ B\beta_1 B...B\beta_k B \rfloor$$

TOMA 6: 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

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

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

TOMA 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 2 *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* GOTO L$\bar{i}$ *ni de la forma* L$\bar{j}$ 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* 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

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

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

$$\begin{array}{l} \text{N}\bar{r} \leftarrow \text{N}\bar{r} + 1 \\ \tilde{\mathcal{P}} \\ \text{N1} \leftarrow 0 \\ \vdots \\ \text{N}\bar{r} \leftarrow 0 \\ \text{P2} \leftarrow \varepsilon \\ \vdots \\ \text{P}\bar{r} \leftarrow \varepsilon \end{array}$$

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 3 *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 2 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 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 $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m}$,

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

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

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

$$\lfloor q_0 B^{k+1} \alpha \rfloor \overset{*}{\vdash} \lfloor q_f B \alpha \rfloor$$

Sea M la maquina dada por el siguiente diagrama

Figura 11

A continuacion veremos que M computa a f. Supongamos que $(\vec{x}, \vec{\alpha}) \in (\omega^n \times \Sigma^{*m}) - D_f$. Deberemos ver que M no termina partiendo de

(*) $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B\alpha_1 B...B\alpha_m B \rfloor$

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

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

lo cual implica (Lema 1) que

(**) M_{sim} no termina partiendo de $\left\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B^{k-n} B\alpha_1 B...B\alpha_m B \right\rfloor$

Ahora notese que si hacemos funcionar a M desde la descripcion instantanea dada en (*), llegaremos (via la copia de M_1 dentro de M) indefectiblemente (ya que M es deterministica) a la siguiente descripcion instantanea

$$\left\lfloor q_2 B \shortmid^{x_1} B...B \shortmid^{x_n} B^{k-n} B\alpha_1 B...B\alpha_m B \right\rfloor$$

Luego entonces (**) nos dice que al seguir trabajando M (ahora via la copia de M_{sim} dentro de M), la maquina M nunca terminara.

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

$$\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B\alpha_1 B...B\alpha_m B \rfloor \overset{*}{\underset{M}{\vdash}} \lfloor q_5 B f(\vec{x}, \vec{\alpha}) \rfloor$$

y que la maquina M se detiene en $\lfloor q_5 B f(\vec{x}, \vec{\alpha}) \rfloor$. La maquina 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 el. Ya que $\mathcal{P}$ computa a f y tiene la propiedad (2) del Lema 2, tenemos que $\mathcal{P}$ termina partiendo de $\|x_1, ..., x_n, \alpha_1, ..., \alpha_m\|$ y llega al estado $\|f(\vec{x}, \vec{\alpha})\|$, o lo que es lo mismo, $\mathcal{P}$ termina partiendo de

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

y llega al estado

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

Pero entonces el Lema 1 nos dice que

(***) $\left\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B^{k-n} B\alpha_1 B...B\alpha_m B \right\rfloor \overset{*}{\underset{M_{sim}}{\vdash}} \left\lfloor q_f B^{k+1} f(\vec{x}, \vec{\alpha}) \right\rfloor$

Como ya lo vimos, si hacemos funcionar a M desde $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B\alpha_1 B...B\alpha_m B \rfloor$, llegaremos (via la copia de M_1 dentro de M) indefectiblemente a la siguiente descripcion instantanea

$$\left\lfloor q_2 B \shortmid^{x_1} B...B \shortmid^{x_n} B^{k-n} B\alpha_1 B...B\alpha_m B \right\rfloor$$

Luego (***) nos dice que, via 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, via 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 ■