Resumen de teoremas para el final de Lenguajes Formales y Computabilidad

Agustín Curto, agucurto95@gmail.com
Francisco Nievas, frannievas@gmail.com

2017

Índice general

0.1. Notación y conceptos basicos . . . 2
- 0.1.1. Funciones Σ-mixtas . . . 2
- 0.1.2. Predicados Σ-mixtos . . . 3
- 0.1.3. Conjuntos Σ-mixtos . . . 4
- 0.1.4. Notacion lambda . . . 4
- 0.1.5. Ordenes naturales sobre Σ^* . . . 6
- 0.1.6. Codificacion de sucesiones infinitas de numeros . . . 9

0.2. Procedimientos efectivos . . . 10
- 0.2.1. Funciones Σ-efectivamente computables . . . 10
- 0.2.2. Conjuntos Σ-efectivamente enumerables . . . 12
- 0.2.3. Conjuntos Σ-efectivamente computables . . . 13

0.3. Funciones Σ-recursivas . . . 14
- 0.3.1. Funciones Σ-recursivas primitivas . . . 14
- 0.3.2. Minimizacion y funciones Σ-recursivas . . . 22
- 0.3.3. Recursion primitiva sobre valores anteriores . . . 25
- 0.3.4. Independencia del alfabeto . . . 25

0.4. El lenguaje $\mathcal{S}^\Sigma$ . . . 28
- 0.4.1. Sintaxis de $\mathcal{S}^\Sigma$ . . . 28

0.5. Maquinas de Turing . . . 46
- 0.5.1. Funciones Σ-Turing computables . . . 50

0.1. Notación y conceptos basicos

Usaremos $\mathbf{N}$ para denotar el conjunto de los números naturales y ω para denotar al conjunto $\mathbf{N}\cup\{0\}$. Dados conjuntos $A_1,...,A_n$ usaremos $A_1\times...\times A_n$ para denotar el producto Cartesiano de $A_1,...,A_n$, es decir el conjunto formado por todas las n-uplas $(a_1,...,a_n)$ tales que $a_1\in A_1,...,a_n\in A_n$. Si $A_1=...=A_n=A$, entonces escribiremos A^n en lugar de $A_1\times...\times A_n$. Usaremos $\Diamond$ para denotar la unica 0-upla. O sea que $A^0=\{\Diamond\}$. Si $A_1,A_2,...$ es una sucesion infinita de conjuntos, entonces usaremos $\bigcup_{i=1}^{\infty}A_i$ o $\bigcup_{i\geq 1}A_i$ para denotar al conjunto

$\{a : a\in A_i, \text{ para algun } i\in\mathbf{N}\}$

Una funcion es un conjunto f de pares ordenados con la siguiente propiedad

- Si $(x,y)\in f$ y $(x,z)\in f$, entonces $y=z$. Dada una funcion f, definamos

$$\begin{aligned} D_f &= \text{dominio de } f=\{x:(x,y)\in f \text{ para algun } y\} \\ I_f &= \text{imagen de } f=\{y:(x,y)\in f \text{ para algun } x\}\end{aligned}$$

Como es usual dado $x\in D_f$, usaremos $f(x)$ para denotar al unico $y\in I_f$ tal que $(x,y)\in f$. Notese que $\varnothing$ es una funcion. Escribiremos $f:S\subseteq A\rightarrow B$ para expresar que f es una funcion tal que $D_f=S\subseteq A$ y $I_f\subseteq B$. Tambien escribiremos $f:A\rightarrow B$ para expresar que f es una funcion tal que $D_f=A$ y $I_f\subseteq B$. Una funcion f es inyectiva cuando no se da que $f(a)=f(b)$ para agun par de lementos distintos a,b. Dada una funcion $f:A\rightarrow B$ diremos que f es suryectiva cuando $I_f=B$. Debe notarse que el concepto de suryectividad depende de un conjunto previamente fijado, B el cual contenga a I_f, no tiene sentido hablar de la suryectividad de una funcion f si no decimos respecto de que conjunto lo es. Dada una funcion $f:A\rightarrow B$ diremos que f es biyectiva cuando f sea inyectiva y suryectiva.

Un alfabeto es un conjunto finito de simbolos. Notese que $\varnothing$ es un alfabeto. Si Σ es un alfabeto no vacio, entonces Σ^* denotara el conjunto de todas las palabras formadas con simbolos de Σ. Por convension definiremos $\varnothing^*=\varnothing$. Usaremos $|\alpha|$ para denotar la longitud de la palabra α. La unica palabra de longitud 0 es denotada con ε. Notese que funciones, n-uplas y palabras son objetos de distinta naturaleza por lo cual $\varnothing$, $\Diamond$ y ε son tres objetos matematicos diferentes.

Si $\alpha_1,...,\alpha_n\in\Sigma^*$, usaremos $\alpha_1...\alpha_n$ para denotar la concatenacion de las palabras $\alpha_1,...,\alpha_n$. Si $\alpha_1=...=\alpha_n=\alpha$, entonces escribiremos α^n en lugar de $\alpha_1...\alpha_n$. Por convencion $\alpha^0=\varepsilon$. Diremos que β es un tramo inicial (propio) de α si hay una palabra γ tal que $\alpha=\beta\gamma$ (y $\beta\notin\{\varepsilon,\alpha\}$). En forma similar se define tramo final (propio).

Dados $i\in\omega$ y $\alpha\in\Sigma^*$ definamos

$$[\alpha]_i=\begin{cases} i\text{-esimo elemento de } \alpha & \text{si } 1\leq i\leq|\alpha| \\ \varepsilon & \text{caso contrario}\end{cases}$$

Para $x,y\in\omega$, definamos $x\dot{-}y=\begin{cases} x-y & \text{si } x\geq y \\ 0 & \text{caso contrario}\end{cases}$

Dados $x,y\in\omega$ diremos que x divide a y cuando haya un $z\in\omega$ tal que $y=z.x$. Escribiremos $x\mid y$ para expresar que x divide a y. Usaremos $\omega^n\times\Sigma^{*m}$ para abreviar la expresion

$\overbrace{\omega\times...\times\omega}^{n}\times\overbrace{\Sigma^*\times...\times\Sigma^*}^{m}$

Por ejemplo, cuando $n=m=0$, tenemos que $\omega^n\times\Sigma^{*m}$ denota el conjunto $\{\Diamond\}$ y si $m=0$, entonces $\omega^n\times\Sigma^{*m}$ denota el conjunto ω^n. Escribiremos $(\vec{x},\vec{\alpha})$ en lugar de $(x_1,...,x_n,\alpha_1,...,\alpha_m)$. Notese que cuando $\Sigma=\varnothing$, entonces $\omega^n\times\Sigma^{*m}=\varnothing$, si $m\geq 1$.

0.1.1. Funciones Σ-mixtas

Una funcion f es llamada Σ-mixta si existen $n,m\geq 0$, tales que $D_f\subseteq\omega^n\times\Sigma^{*m}$ y ya sea $I_f\subseteq\omega$ o $I_f\subseteq\Sigma^*$. Dada una funcion Σ-mixta $f:S\subseteq\omega^n\times\Sigma^{*m}\rightarrow O$, con $O\in\{\omega,\Sigma^*\}$, diremos que f es Σ- total cuando $D_f=\omega^n\times\Sigma^{*m}$. Cabe destacar que si $\Sigma\subseteq\Gamma$, entonces toda función Σ -mixta es Γ-mixta. Sin embargo una función puede ser Σ-total y no ser Γ-total, cuando $\Sigma\subseteq\Gamma$.

Dado que $\varnothing^* = \varnothing$, una función f es $\varnothing$-mixta si y solo si existe $n \geq 0$, tal que $D_f \subseteq \omega^n$ y $I_f \subseteq \omega$.

A continuación daremos algunas funciones Σ-mixtas básicas las cuales serán frecuentemente usadas.

La función sucesor es definida por

$$\begin{array}{rcl} Suc : \omega & \rightarrow & \omega \\ n & \rightarrow & n+1 \end{array}$$

La función predecesor es definida por

$$\begin{array}{rcl} Pred : \mathbf{N} & \rightarrow & \omega \\ n & \rightarrow & n-1 \end{array}$$

Para cada $a \in \Sigma$, definamos

$$\begin{array}{rcl} d_a : \Sigma^* & \rightarrow & \Sigma^* \\ \alpha & \rightarrow & \alpha a \end{array}$$

Para $n, m \in \omega$ y $n \geq i \geq 1$, definamos

$$\begin{array}{rcl} p_i^{n,m} : \omega^n \times \Sigma^{*m} & \rightarrow & \omega \\ (\vec{x}, \vec{\alpha}) & \rightarrow & x_i \end{array}$$

Para $n, m \in \omega$ y $n + m \geq i \geq n+1$, definamos

$$\begin{array}{rcl} p_i^{n,m} : \omega^n \times \Sigma^{*m} & \rightarrow & \Sigma^* \\ (\vec{x}, \vec{\alpha}) & \rightarrow & \alpha_{i-n} \end{array}$$

Las funciones $p_i^{n,m}$ son llamadas proyecciones. Para $n, m, k \in \omega$, y $\alpha \in \Sigma^*$, definamos

$$\begin{array}{rcl} C_k^{n,m} : \omega^n \times \Sigma^{*m} & \rightarrow & \omega \\ (\vec{x}, \vec{\alpha}) & \rightarrow & k \end{array} \qquad \begin{array}{rcl} C_\alpha^{n,m} : \omega^n \times \Sigma^{*m} & \rightarrow & \Sigma^* \\ (\vec{x}, \vec{\alpha}) & \rightarrow & \alpha \end{array}$$

Notese que $C_k^{0,0} : \{\Diamond\} \rightarrow \{k\}$ y que $p_i^{n,0}, C_k^{n,0} : \omega^n \rightarrow \omega$.

Definamos

$$\begin{array}{rcl} pr : \mathbf{N} & \rightarrow & \omega \\ n & \rightarrow & n\text{-esimo numero primo} \end{array}$$

Notese que $pr(1) = 2$, $pr(2) = 3$, $pr(3) = 5$, etc.

0.1.2. Predicados Σ-mixtos

Un predicado Σ-mixto es una funcion f la cual es Σ-mixta y ademas cumple que $I_f \subseteq \{0, 1\}$. Por ejemplo

$$\begin{array}{rcl} \omega \times \omega & \rightarrow & \omega \\ (x, y) & \rightarrow & \left\{ \begin{array}{l} 1 \text{ si } x = y \\ 0 \text{ si } x \neq y \end{array} \right. \end{array} \qquad \begin{array}{rcl} \omega \times \Sigma^* & \rightarrow & \omega \\ (x, \alpha) & \rightarrow & \left\{ \begin{array}{l} 1 \text{ si } x = |\alpha| \\ 0 \text{ si } x \neq |\alpha| \end{array} \right. \end{array}$$

Operaciones logicas entre predicados

Dados predicados $P : S \subseteq \omega^n \times \Sigma^{*m} \rightarrow \{0, 1\}$ y $Q : S \subseteq \omega^n \times \Sigma^{*m} \rightarrow \{0, 1\}$, con el mismo dominio, definamos nuevos predicados $(P \vee Q)$, $(P \wedge Q)$ y $\neg P$ de la siguiente manera

$$\begin{array}{rcl} (P \vee Q) : S & \rightarrow & \omega \\ (\vec{x}, \vec{\alpha}) & \rightarrow & \left\{ \begin{array}{ll} 1 & \text{si } P(\vec{x}, \vec{\alpha}) = 1 \text{ o } Q(\vec{x}, \vec{\alpha}) = 1 \\ 0 & \text{caso contrario} \end{array} \right. \end{array}$$

$$\begin{array}{rcl} (P \wedge Q) : S & \rightarrow & \omega \\ (\vec{x}, \vec{\alpha}) & \rightarrow & \left\{ \begin{array}{ll} 1 & \text{si } P(\vec{x}, \vec{\alpha}) = 1 \text{ y } Q(\vec{x}, \vec{\alpha}) = 1 \\ 0 & \text{caso contrario} \end{array} \right. \end{array}$$

$$\begin{array}{rcl} \neg P : S & \rightarrow & \omega \\ (\vec{x}, \vec{\alpha}) & \rightarrow & \left\{ \begin{array}{ll} 1 & \text{si } P(\vec{x}, \vec{\alpha}) = 0 \\ 0 & \text{si } P(\vec{x}, \vec{\alpha}) = 1 \end{array} \right. \end{array}$$

0.1.3. Conjuntos Σ-mixtos

Un conjunto S es llamado Σ-mixto si $S \subseteq \omega^n \times \Sigma^{*m}$, con $n, m \geq 0$. Notese que S es $\varnothing$ -mixto sii existe un $n \geq 0$, tal que $S \subseteq \omega^n$.

Dado $S \subseteq \omega^n \times \Sigma^{*m}$ usaremos χ_S para denotar la funcion

$$\begin{array}{rcl} \chi_S : \omega^n \times \Sigma^{*m} & \rightarrow & \omega \\ (\vec{x}, \vec{\alpha}) & \rightarrow & \left\{ \begin{array}{l} 1 \text{ si } (\vec{x}, \vec{\alpha}) \in S \\ 0 \text{ si } (\vec{x}, \vec{\alpha}) \notin S \end{array} \right. \end{array}$$

Llamaremos a χ_S la funcion caracteristica de S. Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado rectangular si es de la forma

$S_1 \times ... \times S_n \times L_1 \times ... \times L_m$,

con $n, m \geq 0$, cada $S_i \subseteq \omega$ y cada $L_i \subseteq \Sigma^*$. El concepto de conjunto rectangular es muy importante en nuestro enfoque. Aunque en general no habra restricciones acerca del dominio de las funciones y predicados, nuestra filosofia sera tratar en lo posible que los dominios de las funciones que utilicemos para hacer nuestro analisis de recursividad de distintos paradigmas, sean rectangulares. Aunque en principio puede pareser que todos los conjuntos son rectangulares, el siguiente lema mostrara cuan ingenua es esta vision. Lema 1 Sea $S \subseteq \omega \times \Sigma^*$. Entonces S es rectangular si y solo si se cumple la siguiente propiedad: R Si $(x, \alpha), (y, \beta) \in S$, entonces $(x, \beta) \in S$ Prueba: Ejercicio. □

Supongamos $\Sigma = \{\#, \&, \%\}$. Notese que podemos usar el lema anterior para probar por ejemplo que los siguientes conjuntos no son rectangulares

- $\{(0, \#\#), (1, \%\%\%)\}$ - $\{(x, \alpha) : |\alpha| = x\}$ Dejamos como ejercicio para el lector enunciar un lema analogo al anterior pero que caracterice cuando $S \subseteq \omega^2 \times \Sigma^{*3}$ es rectangular.

0.1.4. Notacion lambda

Usaremos la notacion lambda de Church en la forma que se explica a continuacion. Supongamos ya hemos fijado un alfabeto Σ y supongamos E es una expresion la cual involucra variables numericas y variables alfabeticas y la cual puede estar definida o no dependiendo de que valores concretos le damos a cada una de dichas variables. Supongamos tambien que cuando E esta definida, produce un valor ya sea numerico o alfabetico. Sea $x_1, ..., x_n, \alpha_1, ..., \alpha_m$ una lista de variables tal que las variables numericas que ocurren en E estan todas contenidas en la lista $x_1, ..., x_n$ y que las alfabeticas lo estan en la lista $\alpha_1, ..., \alpha_m$. Entonces

$\lambda x_1...x_n\alpha_1...\alpha_m [E]$

denotara la funcion definida por: El dominio de $\lambda x_1...x_n\alpha_1...\alpha_m [E]$ es el conjunto de las $(n+m)$-uplas $(k_1, ..., k_n, \beta_1, ..., \beta_m) \in \omega^n \times \Sigma^{*m}$ tales que E esta definida cuando le asignamos a cada x_i el valor k_i y a cada α_i el valor β_i. $\lambda x_1...x_n\alpha_1...\alpha_m [E] (k_1, ..., k_n, \beta_1, ..., \beta_m) =$ valor que asume E cuando le asignamos a cada x_i el valor k_i y a cada α_i el valor β_i. Ejemplos:

(a) $\lambda\alpha\beta [\alpha\beta]$ es la funcion $\begin{array}{rcl} \Sigma^* \times \Sigma^* & \rightarrow & \Sigma^* \\ (\alpha, \beta) & \rightarrow & \alpha\beta \end{array}$

(b) $\lambda\beta\alpha [\alpha\beta]$ es la funcion $\begin{array}{rcl} \Sigma^* \times \Sigma^* & \rightarrow & \Sigma^* \\ (\alpha, \beta) & \rightarrow & \beta\alpha \end{array}$

(c) $\lambda xy\alpha\beta [Pred(|\alpha|) + Pred(y)]$ es la funcion $\begin{array}{rcl} \{(x, y, \alpha, \beta) \in \omega^2 \times \Sigma^{*2} : |\alpha| \geq 1 \text{ y } y \geq 1\} & \rightarrow & \omega \\ (x, y, \alpha, \beta) & \rightarrow & Pred(|\alpha \end{array}$

(d) Si la expresion E, cuando esta definida, toma valores Booleanos 0 o 1, entonces la funcion $\lambda x_1...x_n\alpha_1...\alpha_m [E]$ es un predicado. Por ejemplo $\lambda xy [x = y]$ es el predicado $\begin{array}{rcl} \omega \times \omega & \rightarrow & \omega \\ (x, y) & \rightarrow & \left\{ \begin{array}{l} 1 \text{ si } x = y \\ 0 \text{ si } x \neq y \end{array} \right. \end{array}$

y $\lambda x\alpha [Pred(x) = |\alpha|]$ es el predicado $\begin{array}{rcl} \mathbf{N} \times \Sigma^* & \rightarrow & \omega \\ (x, \alpha) & \rightarrow & \left\{ \begin{array}{l} 1 \text{ si } Pred(x) = |\alpha| \\ 0 \text{ si } Pred(x) \neq |\alpha| \end{array} \right. \end{array}$

(e) No toda variable de la lista $x_1, ..., x_n, \alpha_1, ..., \alpha_m$ debe ocurrir en E. Por ejemplo $\lambda xy\alpha\,[Pred(y)]$ es la funcion $\begin{array}{rcl} \{(x,y,\alpha) \in \omega^2 \times \Sigma^* : y \geq 1\} & \rightarrow & \omega \\ (x,y,\alpha) & \rightarrow & Pred(y) \end{array}$

(f) Notar que para $S \subseteq \omega^n \times \Sigma^{*m}$ se tiene que $\chi_S = \lambda x_1...x_n\alpha_1...\alpha_m\,[(\vec{x},\vec{\alpha}) \in S]$. Notar que la notacion λ depende del alfabeto Σ previamente fijado. Cuando haya varios alfabetos bajo consideracion, en general resultara claro del contexto con respecto a cual de ellos se usa la notacion λ.

0.1.5. Ordenes naturales sobre Σ^*

Sea A un conjunto no vacio cualquiera. Una relacion binaria $<$ sobre A sera llamada un orden total estricto sobre A si se cumplen las siguientes condiciones:

(1) Para todo $a \in A$, no se da que $a < a$ (2) Para todo $a, b \in A$, si $a \neq b$, entonces $a < b$ o $b < a$ (3) Para todo $a, b, c \in A$, si $a < b$ y $b < c$, entonces $a < c$. Sea Σ un alfabeto no vacio y supongamos $<$ es un orden total estricto sobre Σ. Supongamos que $\Sigma = \{a_1, ..., a_n\}$, con $a_1 < a_2 < ... < a_n$. Podemos extender $<$ a un orden total estricto sobre Σ^*, de la siguiente manera

- $\alpha < \beta$, siempre que $|\alpha| < |\beta|$ - $\alpha a_i \beta < \alpha a_j \gamma$, siempre que $|\beta| = |\gamma|$ y $i < j$ Lema 2 La relacion $<$ es un orden total estricto sobre Σ^*. Prueba: Facil. □

Lema 3 La funcion $s^< : \Sigma^* \to \Sigma^*$, definida recursivamente de la siguiente manera:
$$\begin{array}{rcl} s^<(\varepsilon) & = & a_1 \\ s^<(\alpha a_i) & = & \alpha a_{i+1},\ i \\ s^<(\alpha a_n) & = & s^<(\alpha) a_1 \end{array}$$

tiene la siguiente propiedad $s^<(\alpha) = \min\{\beta \in \Sigma^* : \alpha < \beta\}$. Prueba: Supongamos que $\alpha < \beta$. Probaremos entonces que $s^<(\alpha) \leq \beta$.

Caso $|\alpha| < |\beta|$.

Se puede ver facilmente que $|\alpha| = |s^<(\alpha)|$ salvo en el caso en que $\alpha \in \{a_n\}^*$, por lo cual solo resta ver el caso $\alpha \in \{a_n\}^*$. Supongamos $\alpha = a_n^{|\alpha|}$. Entonces $s^<(\alpha) = a_1^{|\alpha|+1}$. Si $|\beta| = |\alpha| + 1$ entonces es facil ver usando 2. de la definicion del orden de Σ^* que $s^<(\alpha) = a_1^{|\alpha|+1} \leq \beta$. Si $|\beta| > |\alpha| + 1$, entonces por 1. de tal definicion tenemos que $s^<(\alpha) = a_1^{|\alpha|+1} < \beta$

Caso $|\alpha| = |\beta|$.

Tenemos entonces que
$$\begin{array}{rcl} \alpha & = & \alpha_1 a_i \gamma_1 \\ \beta & = & \alpha_1 a_j \gamma_2 \end{array}$$
con $i < j$ y $|\gamma_1| = |\gamma_2|$. Si $\gamma_1 = \gamma_2 = \varepsilon$ entonces es claro que $s^<(\alpha) \leq \beta$. El caso en el que γ_1 termina con a_l para algun $l < n$ es facil. Veamos el caso en que $\gamma_1 = a_n^k$ con $k \geq 1$. Tenemos que
$$\begin{array}{rcl} s^<(\alpha) & = & s^<(\alpha_1 a_i a_n^k) \\ & = & s^<(\alpha_1 a_i a_n^{k-1}) a_1 \\ \vdots & & \vdots \\ & = & s^<(\alpha_1 a_i) a_1^k \\ & = & \alpha_1 a_{i+1} a_1^k \\ & \leq & \alpha_1 a_j \gamma_2 = \beta \end{array}$$

Supongamos finalmente que $\gamma_1 = \rho_1 a_l a_n^k$ con $k \geq 1$ y $l < n$. Tenemos que
$$\begin{array}{rcl} s^<(\alpha) & = & s^<(\alpha_1 a_i \rho_1 a_l a_n^k) \\ & = & s^<(\alpha_1 a_i \rho_1 a_l a_n^{k-1}) a_1 \\ \vdots & & \vdots \\ & = & s^<(\alpha_1 a_i \rho_1 a_l) a_1^k \\ & = & \alpha_1 a_i \rho_1 a_{l+1} a_1^k \\ & \leq & \beta. \end{array}$$

Para completar nuestra demostracion debemos probar que $\alpha < s^<(\alpha)$, para cada $\alpha \in \Sigma^*$. Dejamos al lector como ejercicio esta prueba la cual puede ser hecha por inducion en $|\alpha|$ usando argumentos parecidos a los usados recien. □ En virtud del lema anterior llamaremos a $s^<$ la funcion sucesor respecto del orden $<$ de Σ^*.

Corolario 4 $s^<$ es inyectiva. Prueba: Supongamos $\alpha \neq \beta$. Ya que el orden de Σ^* es total podemos suponer sin perdida de generalidad que $\alpha < \beta$. Por el lema anterior tenemos que $s^<(\alpha) \leq \beta < s^<(\beta)$ y ya que $<$ es transitiva obtenemos que $s^<(\alpha) < s^<(\beta)$, lo cual nos dice $s^<(\alpha) \neq s^<(\beta)$. □

Lema 5 Se tiene que (1) $\varepsilon \neq s^<(\alpha)$, para cada $\alpha \in \Sigma^*$ (2) Si $\alpha \neq \varepsilon$, entonces $\alpha = s^<(\beta)$ para algun β (3) Si $S \subseteq \Sigma^*$ es no vacio, entonces existe $\alpha \in S$ tal que $\alpha < \beta$, para cada $\beta \in S - \{\alpha\}$. Prueba: (1) y (2) son dejadas al lector. Probaremos (3). Sea $k = \min\{|\alpha| : \alpha \in S\}$. Notese que hay una cantidad finita de palabras de S con longitud igual a k y que la menor de ellas es justamente la menor palabra de S. $\square$

Definamos recursivamente la funcion $*^<: \omega \to \Sigma^*$ de la siguiente manera

$$\begin{array}{rcl} *^<(0) & = & \varepsilon \\ *^<(x+1) & = & s^<(*^<(x)) \end{array}$$

Lema 6 Tenemos que $\Sigma^* = \{*^<(0), *^<(1), ...\}$ Mas aun la funcion $*^<$ es biyectiva. Prueba: Supongamos $*^<(x) = *^<(y)$ con $x > y$. Note que $y \neq 0$ ya que ε no es el sucesor de ninguna palabra. O sea que $s^<(*^<(x-1)) = s^<(*^<(y-1))$ lo cual ya que $*^<$ es inyectiva nos dice que $*^<(x-1) = *^<(y-1)$. Iterando este razonamiento llegamos a que $*^<(z) = *^<(0) = \varepsilon$ para algun $z > 0$, lo cual es absurdo.

Veamos que $*^<$ es sobre. Supongamos no lo es, es decir supongamos que $\Sigma^* - I_{*^<} \neq \varnothing$. Por (3) del lema anterior $\Sigma^* - I_{*^<}$ tiene un menor elemento α. Ya que $\alpha \neq \varepsilon$, tenemos que $\alpha = s^<(\beta)$, para algun β. Ya que $\beta < \alpha$ tenemos que $\beta \notin \Sigma^* - I_{*^<}$, es decir que $\beta = *^<(x)$, para algun $x \in \omega$. Esto nos dice que $\alpha = s^<(*^<(x))$, lo cual por la definicion de $*^<$ nos dice que $\alpha = *^<(x+1)$. Pero esto es absurdo ya que $\alpha \notin I_{*^<}$. $\square$

Lema 7 Sea $n \geq 1$ fijo. Entonces cada $x \geq 1$ se escribe en forma unica de la siguiente manera: $x = i_k n^k + i_{k-1} n^{k-1} + ... + i_0 n^0$, con $k \geq 0$ y $1 \leq i_k, i_{k-1}, ..., i_0 \leq n$. Prueba: Primero la unicidad. Supongamos que

$$i_k n^k + i_{k-1} n^{k-1} + ... + i_0 n^0 = j_m n^m + j_{m-1} n^{m-1} + ... + j_0 n^0$$

con $k, m \geq 0$ y $1 \leq i_k, i_{k-1}, ..., i_0, j_m, ..., j_0 \leq n$. Supongamos $k < m$. Llegaremos a un absurdo. Notese que

$$\begin{array}{rcl} i_k n^k + i_{k-1} n^{k-1} + ... + i_0 n^0 & \leq & n.n^k + n.n^{k-1} + ... + n.n^0 \\ & \leq & n^{k+1} + n^k + ... + n^1 \\ & < & n^{k+1} + n^k + ... + n^1 + n^0 \\ & \leq & n^m + n^{m-1} + ... + n^0 \\ & \leq & j_m n^m + j_{m-1} n^{m-1} + ... + j_0 n^0 \end{array}$$

lo cual contradice la primera igualdad. Probaremos por induccion en x que

(1) Existen $k \geq 0$ y $i_k, i_{k-1}, ..., i_0 \in \{1, ..., n\}$ tales que $x = i_k n^k + i_{k-1} n^{k-1} + ... + i_0 n^0$

El caso $x = 1$ es trivial. Supongamos (1) vale para x, probaremos que vale para $x + 1$. Hay varios casos:

Caso $i_0 < n$. Entonces

$$\begin{array}{rl} x+1 & = \left(i_k n^k + i_{k-1} n^{k-1} + ... + i_0 n^0\right) + 1 \\ & = i_k n^k + i_{k-1} n^{k-1} + ... + (i_0 + 1) n^0 \end{array}$$

Caso $i_k = i_{k-1} = ... = i_0 = n$. Tenemos que

$$\begin{array}{rl} x+1 & = \left(i_k n^k + i_{k-1} n^{k-1} + ... + i_0 n^0\right) + 1 \\ & = \left(n n^k + n n^{k-1} + ... + n n^0\right) + 1 \\ & = 1 n^{k+1} + 1 n^k + ... + 1 n^1 + 1 n^0 \end{array}$$

Caso $i_0 = i_1 = ... = i_h = n,\ i_{h+1} \neq n$, para algun $0 \leq h < k$. Tenemos

$$\begin{array}{rl} x+1 & = \left(i_k n^k + ... + i_{h+2} n^{h+2} + i_{h+1} n^{h+1} + n n^h + ... + n n^0\right) + 1 \\ & = \left(i_k n^k + ... + i_{h+2} n^{h+2} + i_{h+1} n^{h+1} + n^{h+1} + n^h + ... + n^1\right) + 1 \\ & = i_k n^k + ... + i_{h+2} n^{h+2} + (i_{h+1} + 1) n^{h+1} + 1 n^h + ... + 1 n^1 + 1 n^0. \end{array}$$

$\square$ Notese que cada $\alpha \in \Sigma^* - \{\varepsilon\}$ se escribe de la forma

$\alpha = a_{i_k} ... a_{i_0}$

donde $k \geq 0$ y $1 \leq i_k, i_{k-1}, ..., i_0 \leq n$. Definamos la funcion $\#^<$ de la siguiente manera

$$\begin{array}{rcl} \#^< : \Sigma^* & \to & \omega \\ \varepsilon & \to & 0 \\ a_{i_k} ... a_{i_0} & \to & i_k n^k + ... + i_0 n^0 \end{array}$$

Notese que el lema anterior nos dice que fijado $n \geq 1$, los numeros naturales estan identificados o en correspondencia biunivoca con las sucesiones finitas de elementos del conjunto $\{1, ..., n\}$. Ya que podemos identificar cada a_i con i el lema anterior nos garantiza que los numero naturales estan en correspondencia biunivoca con las palabras no nulas del alfabeto Σ. Es decir que hemos probado que Lema 8 La funcion $\#^<$ es biyectiva Concluimos la seccion con la siguiente observacion

Lema 9 Las funciones $\#^<$ y $*^<$ son una inversa de la otra. Prueba: Probaremos por induccion en x que para cada $x \in \omega$, se tiene que $\#^<(*^<(x)) = x$. El caso $x = 0$ es trivial. Supongamos que $\#^<(*^<(x)) = x$, veremos entonces que $\#^<(*^<(x+1)) = x+1$. Sean $k \geq 0$ y $i_k, ..., i_0$ tales que $*^<(x) = a_{i_0}...a_{i_0}$. Ya que $\#^<(*^<(x)) = x$ tenemos que $x = i_k n^k + ... + i_0 n^0$. Hay varios casos

Caso $i_0 < n$. Entonces $*^<(x+1) = s^<(*^<(x)) = a_{i_k}...a_{i_0+1}$ por lo cual

$$\begin{aligned} \#^<(*^<(x+1)) &= i_k n^k + i_{k-1} n^{k-1} + ... + (i_0+1) n^0 \\ &= \left(i_k n^k + i_{k-1} n^{k-1} + ... + i_0 n^0\right) + 1 \\ &= x + 1 \end{aligned}$$

Caso $i_k = i_{k-1} = ... = i_0 = n$. Entonces $*^<(x+1) = s^<(*^<(x)) = a_1^{k+2}$ por lo cual

$$\begin{aligned} \#^<(*^<(x+1)) &= 1n^{k+1} + 1n^k + ... + 1n^1 + 1n^0 \\ &= \left(nn^k + nn^{k-1} + ... + nn^0\right) + 1 \\ &= x + 1 \end{aligned}$$

Caso $i_0 = i_1 = ... = i_h = n, \;\; i_{h+1} \neq n$, para algun $0 \leq h < k$. Entonces $*^<(x+1) = s^<(*^<(x)) = a_{i_k}...a_{i_{h+2}} a_{i_{h+1}+1} a_1...a_1$ por lo cual

$$\begin{aligned} \#^<(*^<(x+1)) &= i_k n^k + ... + i_{h+2} n^{h+2} + (i_{h+1}+1) n^{h+1} + 1n^h + ... + 1n^1 + 1n^0 \\ &= \left(i_k n^k + ... + i_{h+2} n^{h+2} + i_{h+1} n^{h+1} + n^{h+1} + n^h + ... + n^1\right) + 1 \\ &= \left(i_k n^k + ... + i_{h+2} n^{h+2} + i_{h+1} n^{h+1} + nn^h + ... + nn^0\right) + 1 \\ &= x + 1 \end{aligned}$$

□

0.1.6. Codificacion de sucesiones infinitas de numeros

Dado $n \in \mathbf{N}$, usaremos $pr(n)$ para denotar el n-esimo numero primo. Notese que $pr(1) = 2$, $pr(2) = 3$, $pr(3) = 5$, etc. Usaremos $\omega^{\mathbf{N}}$ para denotar el conjunto de todas las sucesiones infinitas de elementos de ω. Es decir

$\omega^{\mathbf{N}} = \{(s_1, s_2, ...) : s_i \in \omega, \text{ para cada } i \geq 1\}$.

Definamos el siguiente subconjunto de $\omega^{\mathbf{N}}$ $\omega^{[\mathbf{N}]} = \Big\{(s_1, s_2, ...) \in \omega^{\mathbf{N}} :$ hay un $n \in \mathbf{N}$ tal que $s_i = 0$, para

Notese que $\omega^{\mathbf{N}} \neq \omega^{[\mathbf{N}]}$, por ejemplo las sucesiones $\begin{array}{l}(10, 20, 30, 40, 50, ...) \\ (1, 0, 1, 0, 1, 0, 1, 0, ...)\end{array}$

no pertenecen a $\omega^{[\mathbf{N}]}$. Notese que $(s_1, s_2, ...) \in \omega^{[\mathbf{N}]}$ si y solo si solo una cantidad finita de coordenadas de $(s_1, s_2, ...)$ son no nulas (i.e. $\{i : s_i \neq 0\}$ es finito) Necesitaremos el siguente lema para probar una version del Teorema Fundamental de la Aritmetica la cual nos sera util para codificar elementos de $\omega^{[\mathbf{N}]}$ con numeros naturales.

Lema 10 Si $p, p_1, ..., p_n$ son numeros primos y p divide a $p_1...p_n$, entonces $p = p_i$, para algun i. Ahora la version del Teorema Fundamental de la Aritmetica.

Teorema 11 Para cada $x \in \mathbf{N}$, hay una unica sucesion $(s_1, s_2, ...) \in \omega^{[\mathbf{N}]}$ tal que $x = \prod_{i=1}^{\infty} pr(i)^{s_i}$ (Notese que $\prod_{i=1}^{\infty} pr(i)^{s_i}$ tiene sentido ya que es un producto que solo tiene una cantidad finita de factores no iguales a 1.) Prueba: Primero probaremos la existencia por induccion en x. Claramente $1 = \prod_{i=1}^{\infty} pr(i)^0$, con lo cual el caso $x = 1$ esta probado. Supongamos la existencia vale para cada y menor que x, veremos que entonces vale para x. Si x es primo, entonces $x = pr(i_0)$ para algun i_0 por lo cual tenemos que $x = \prod_{i=1}^{\infty} pr(i)^{s_i}$, tomando $s_i = 0$ si $i \neq i_0$ y $s_{i_0} = 1$. Si x no es primo, entonces $x = y_1.y_2$ con $y_1, y_2 < x$. Por hipotesis inductiva tenemos que hay $(s_1, s_2, ...), (t_1, t_2, ...) \in \omega^{[\mathbf{N}]}$ tales que $y_1 = \prod_{i=1}^{\infty} pr(i)^{s_i}$ y $y_2 = \prod_{i=1}^{\infty} pr(i)^{t_i}$. Tenemos entonces que $x = \prod_{i=1}^{\infty} pr(i)^{s_i + t_i}$ lo cual concluye la prueba de la existencia.

Veamos ahora la unicidad. Suponganos que

$\prod_{i=1}^{\infty} pr(i)^{s_i} = \prod_{i=1}^{\infty} pr(i)^{t_i}$

Si $s_i > t_i$ entonces dividiendo ambos miembros por $pr(i)^{t_i}$ obtenemos que $pr(i)$ divide a un producto de primos todos distintos de el, lo cual es absurdo por el lema anterior. Analogamente llegamos a un absurdo si suponemos que $t_i > s_i$, lo cual nos dice que $s_i = t_i$, para cada $i \in \mathbf{N}$ $\square$ Notese que razonando con el lema anterior, podemos probar que si $x = \prod_{i=1}^{\infty} pr(i)^{s_i}$, entonces $s_i = \max_t (pr(i)^t \text{ divide a } x)$, para cada $i \in \mathbf{N}$. Definamos para $x, i \in \mathbf{N}$,

$(x)_i = \max\limits_t \Big(pr(i)^t \text{ divide a } x\Big)$.

Dada una sucesion $(s_1, s_2, ...) \in \omega^{[\mathbf{N}]}$ definamos $\langle s_1, s_2, ...\rangle = \prod_{i=1}^{\infty} pr(i)^{s_i}$

Tenemos entonces el siguiente Lema 12 Las funciones $\begin{array}{lll}\mathbf{N} & \rightarrow & \omega^{[\mathbf{N}]} \\ x & \rightarrow & ((x)_1, (x)_2, ...)\end{array}$ $\begin{array}{ll}\omega^{[\mathbf{N}]} & \rightarrow \\ (s_1, s_2, ...) & \rightarrow\end{array}$

son biyecciones una inversa de la otra. Prueba: Notese que para cada $x \in \mathbf{N}$, tenemos que $\langle (x)_1, (x)_2, ...\rangle = x$. Ademas para cada $(s_1, s_2, ...) \in \omega^{[\mathbf{N}]}$, tenemos que $((\langle s_1, s_2, ...\rangle)_1, (\langle s_1, s_2, ...\rangle)_2, ...) = (s_1, s_2, ...)$. Es claro que lo anterior garantiza que los mapeos en cuestion son uno inversa del otro $\square$

Para $x \in \mathbf{N}$ definamos:

$Lt(x) = \begin{cases} \max_i (x)_i \neq 0 & \text{si } x \neq 1 \\ 0 & \text{si } x = 1 \end{cases}$

Se tienen las siguientes propiedades basicas Lema 13 Para cada $x \in \mathbf{N}$: $Lt(x) = 0$ sii $x = 1$ $x = \prod_{i=1}^{Lt(x)} pr(i)^{(x)_i}$ Cabe destacar entonces que la funcion $\lambda ix[(x)_i]$ tiene dominio igual a $\mathbf{N}^2$ y la funcion $\lambda ix[Lt(x)]$ tiene dominio igual a $\mathbf{N}$.

0.2. Procedimientos efectivos

Un concepto importante en ciencias de la computacion es el de procedimiento o metodo para realizar alguna tarea determinada. Nos interesan los procedimientos que estan definidos en forma precisa e inambigua, es decir aquellos en los cuales en cada etapa a seguir, la tarea a realizar esta objetivamente descripta.

Tambien podemos encontrar en procedimientos rigurosamente definidos la propiedad de no terminacion. Es decir puede suceder que para ciertos datos de entrada, la ejecucion del procedimiento nunca llegue a producir un resultado o dato de salida sino que a medida que se vayan realizando las instrucciones o tareas, siempre el procedimiento direccione a realizar otra tarea especifica y asi sucesivamente.

Cabe destacar que los procedimientos tambien deben ser repetibles, en el sentido de que si realizamos un procedimiento dos veces con el mismo dato de entrada, entonces ya sea en cada una de estas realizaciones el procedimiento no termina o en las dos termina y da como resultado el mismo dato de salida.

Una caracteristica de los procedimientos que surgen en la tarea cientifica es que hay un conjunto de datos de entrada, es decir, el conjunto de objetos a partir de los cuales puede comenzar a realizarse el procedimiento. Cabe destacar que para ciertos elementos de dicho conjunto puede pasar que el proceimiento no termine partiendo de ellos. Tambien en los procedimientos que surgen en la tarea cientifica tenemos un conjunto de datos de salida, es decir el conjunto de todos los datos que el procedimiento dara como salida al terminar partiendo de los distintos datos de entrada.

Otro aspecto muy importante a considerar es que un procedimiento puede tener pasos a seguir los cuales sean realizables solo en un sentido puramente teorico. Por ejemplo, un procedimiento puede tener una instruccion como la que se muestra a continuacion:

- si el polinomio $ax^5 + bx^4 + 421$ tiene una raiz racional, entonces realizar la tarea descripta en A, en caso contrario realizar la tarea descripta en B (a, b son datos calculados previamente). Como puede notarse mas alla de este aspecto teorico de la instruccion, su descripcion es clara y objetiva, pero en principio no es claro que se pueda ejecutar dicha instruccion en un sentido efectivo a los fines de seguir realizando las siguientes instrucciones.

Un procedimiento sera llamado efectivo cuando cada paso del mismo sea efectivamente realizable.

0.2.1. Funciones Σ-efectivamente computables

Trabajaremos con procedimientos efectivos en los cuales el conjunto de datos de entrada es $\omega^n \times \Sigma^{*m}$ para algunos $n, m \geq 0$ y el conjunto de datos de salida esta contenido en $\omega^k \times \Sigma^{*l}$ para algunos $k, l \geq 0$. Tambien supondremos que los elementos de ω que intervienen en los datos de entrada y de salida estaran representados en notacion decimal.

Una funcion Σ-mixta, $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$, sera llamada Σ-efectivamente computable si hay un procedimiento efectivo $\mathbb{P}$ con las siguientes caracteristicas:

- El conjunto de datos de entrada de $\mathbb{P}$ es $\omega^n \times \Sigma^{*m}$ - El conjunto de datos de salida esta contenido en ω. - Si $(\vec{x}, \vec{\alpha}) \in D_f$ y corremos $\mathbb{P}$ desde $(\vec{x}, \vec{\alpha})$, entonces $\mathbb{P}$ termina y da como dato de salida $f(\vec{x}, \vec{\alpha})$. - Si $(\vec{x}, \vec{\alpha}) \in (\omega^n \times \Sigma^{*m}) - D_f$, entonces $\mathbb{P}$ no termina partiendo de $(\vec{x}, \vec{\alpha})$ En forma analoga se define cuando una funcion $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \Sigma^*$ es Σ -efectivamente computable. En ambos casos diremos que la funcion f es computada por $\mathbb{P}$.

Quisas el procedimiento mas famoso de la matematica es aquel que se enseñ a en los colegios para sumar dos numeros naturales expresados en notacion decimal. Es decir que la funcion $\lambda xy\,[x + y]$ es Σ -efectivamente computable, cualquiera sea el alfabeto Σ. Tambien las funciones $\lambda xy\,[x.y]\,, \lambda xy\,[x^y]$ son Σ-efectivamente computables via los procedimientos clasicos enseñ ados en la escuela primaria. Veamos algunos ejemplos mas

Ejemplos: Consideremos $\Sigma = \{\%, \&\}$.

(a) La funcion $C_3^{1,2}$ es Σ-efectivamente computable ya que el siguiente procedimiento $\mathbb{P}$ con conjunto de datos de entrada $\omega \times \Sigma^{*2}$ la computa:

- Independientemente de quien sea el dato de entrada $(x_1, \alpha_1, \alpha_2)$, terminar y dar como salida el numero 3

(b) La funcion $p_3^{2,3}$ es Σ-efectivamente computable ya que el siguiente procedimiento la computa:

- Dado el dato de entrada $(x_1, x_2, \alpha_1, \alpha_2, \alpha_3)$, terminar y dar como salida la palabra α_1

(c) $Pred$ es Σ-efectivamente computable. El siguiente procedimiento (con conjunto de datos de entrada igual a ω) computa a $Pred$:

Dado como dato de entrada un elemento $x \in \omega$, realizar lo siguiente:

Etapa 1

Si $x = 0$, entonces ir a Etapa 3, en caso contrario ir a Etapa 2.

Etapa 2

Si $x \neq 0$, entonces detenerse y dar como salida el valor $x - 1$.

Etapa 3

Si $x = 0$, entonces ir a Etapa 1.

(d) Si $<$ es el orden total estricto sobre Σ dado por $\& < \%$, entonces ya que $s^< : \Sigma^* \to \Sigma^*$ es definida por

$$\begin{array}{rcl} s^<(\varepsilon) & = & \& \\ s^<(\alpha\&) & = & \alpha\% \\ s^<(\alpha\%) & = & s^<(\alpha)\& \end{array}$$

tenemos que $s^<$ es Σ-efectivamente computable. Por ejemplo el siguiente procedimiento la computa. Tal como en los lenguajes de programacion, usaremos variables y asignaciones para diseñar el procedimiento. Etapa 1: Hacer las siguientes asignaciones

$$\begin{array}{lcl} A & \leftarrow & \alpha \\ B & \leftarrow & \varepsilon \\ F & \leftarrow & \& \end{array}$$

e ir a Etapa 2. Etapa 2: Si A comiensa con $\&$, entonces hacer las siguientes asignaciones

$$\begin{array}{lcl} A & \leftarrow & \text{resultado de remover el 1er simbolo de } A \\ B & \leftarrow & B\& \\ F & \leftarrow & B\% \end{array}$$

e ir a la Etapa 2. En caso contrario ir a la Etapa 3. Etapa 3: Si A comiensa con $\%$, entonces hacer las siguientes asignaciones

$$\begin{array}{lcl} A & \leftarrow & \text{resultado de remover el 1er simbolo de } A \\ B & \leftarrow & B\% \\ F & \leftarrow & F\% \end{array}$$

e ir a la Etapa 2. En caso contrario ir a la Etapa 4. Etapa 4: Si A es ε entonces dar como salida F

(e) Usando que $s^<$ es Σ-efectivamente computable podemos ver que $*^< : \omega \to \Sigma^*$ tambien lo es ya que $*^<$ es definida con las recursiones

$$\begin{array}{rcl} *^<(0) & = & \varepsilon \\ *^<(x+1) & = & s^<(*^<(x)) \end{array}$$

Dejamos como ejercico para el lector diseñar procedimientos efectivos que computen las funciones:

- $\lambda xy[x \text{ divide a } y]$
- $\lambda x[pr(x)]$

. $\lambda ix[(x)_i]$

0.2.2. Conjuntos Σ-efectivamente enumerables

Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ sera llamado Σ-efectivamente enumerable cuando sea vacio o haya un procedimiento efectivo $\mathbb{P}$ con las siguientes caracteristicas:

- El conjunto de datos de entrada de $\mathbb{P}$ es ω y $\mathbb{P}$ termina para cada dato de entrada $x \in \omega$ - El conjunto de datos de salida de $\mathbb{P}$ es S. En tal caso diremos que el procedimiento $\mathbb{P}$ enumera a S . Dicho de otra forma $\mathbb{P}$ debe ser un procedimiento efectivo que para los datos de entrada $0, 1, 2, 3, ...$, termine y produzca datos de salida $e_0, e_1, e_2, e_3, ...$ tales que $S = \{e_0, e_1, e_2, ...\}$.

Ejemplos: (a) El conjunto $S = \{x \in \omega : x \text{ es par}\}$ es Σ-efctivamente enumerable, cualesquiera sea Σ. El siguiente procedimiento enumera a S:

- Calcular $2x$, darlo como dato de salida y terminar.

(b) Un procedimiento que enumera $\omega \times \omega$ es el siguiente:

Etapa 1

Si $x = 0$, dar como salida el par $(0, 0)$ y terminar. Si $x \neq 0$, calcular $x_1 = (x)_1$ y $x_2 = (x)_2$.

Etapa 2

Dar como dato de salida el par (x_1, x_2) y terminar

Como puede notarse el procedimiento es efectivo y si tomamos un par cualquiera $(a, b) \in \omega \times \omega$, el procedimiento lo dara como dato de salida para la entrada $x = 2^a 3^b$.

(c) Veamos que $\omega^2 \times \Sigma^{*3}$ es Σ -efectivamente enumerable cualquiera sea el alfabeto Σ. Sea $<$ un orden total estricto para el alfabeto Σ. Utilisando el orden $<$ podemos diseñar el siguiente procedimiento para enumerar $\omega^2 \times \Sigma^{*3}$:

Etapa 1

Si $x = 0$, dar como salida $(0, 0, \varepsilon, \varepsilon, \varepsilon)$ y terminar. Si $x \neq 0$, calcular

$x_1 = (x)_1$

$x_2 = (x)_2$

$\alpha_1 = *^<((x)_3)$

$\alpha_2 = *^<((x)_4)$

$\alpha_3 = *^<((x)_5)$

Etapa 2

Dar como dato de salida la 5-upla $(x_1, x_2, \alpha_1, \alpha_2, \alpha_3)$.

Lema 14 Sean $S_1, S_2 \subseteq \omega^n \times \Sigma^{*m}$ conjuntos Σ-efectivamente enumerables. Entonces $S_1 \cup S_2$ y $S_1 \cap S_2$ son Σ-efectivamente enumerables. Prueba: El caso en el que alguno de los conjuntos es vacio es trivial. Supongamos que ambos conjuntos son no vacios y sean $\mathbb{P}_1$ y $\mathbb{P}_2$ procedimientos que enumeran a S_1 y S_2. El siguiente procedimiento enumera al conjunto $S_1 \cup S_2$:

- Si x es par realizar $\mathbb{P}_1$ partiendo de $x/2$ y dar el elemento de S_1 obtenido como salida. Si x es impar realizar $\mathbb{P}_2$ partiendo de $(x-1)/2$ y dar el elemento de S_2 obtenido como salida. Veamos ahora que $S_1 \cap S_2$ es Σ-efectivamente enumerable. Si $S_1 \cap S_2 = \varnothing$ entonces no hay nada que probar. Supongamos entonces que $S_1 \cap S_2$ en no vacio. Sea z_0 un elemento fijo de $S_1 \cap S_2$. Sea $\mathbb{P}$ un procedimiento efectivo el cual enumere a $\omega \times \omega$ (ver el ejemplo de mas arriba). Un procedimiento que enumera a $S_1 \cap S_2$ es el siguiente

Etapa 1

Realizar $\mathbb{P}$ con dato de entrada x, para obtener un par $(x_1, x_2) \in \omega \times \omega$.

Etapa 2

Realizar $\mathbb{P}_1$ con dato de entrada x_1 para obtener un elemento $z_1 \in S_1$

Etapa 3

Realizar $\mathbb{P}_2$ con dato de entrada x_2 para obtener un elemento $z_2 \in S_2$

Etapa 4

Si $z_1 = z_2$, entonces dar como dato de salida z_1. En caso contrario dar como dato de salida z_0. $\square$

0.2.3. Conjuntos Σ-efectivamente computables

Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ sera llamado Σ-efectivamente computable cuando haya un procedimiento efectivo $\mathbb{P}$ con las siguientes caracteristicas:

- El conjunto de datos de entrada de $\mathbb{P}$ es $\omega^n \times \Sigma^{*m}$, siempre termina y da como dato de salida un elemento de $\{0,1\}$. - Dado $(\vec{x},\vec{\alpha}) \in \omega^n \times \Sigma^{*m}$, $\mathbb{P}$ da como salida al numero 1 si $(\vec{x},\vec{\alpha}) \in S$ y al numero 0 si $(\vec{x},\vec{\alpha}) \notin S$. Cabe destacar que un conjunto S es Σ-efectivamente computable sii el predicado $\chi_S = \lambda x_1...x_n\alpha_1...\alpha_m\left[(\vec{x},\vec{\alpha}) \in S\right]$ es Σ-efectivamente computable. Un hecho intuitivamente claro es el siguiente

Lema 15 Si $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ -efectivamente computable entonces S es Σ-efectivamente enumerable. Prueba: Supongamos $S \neq \varnothing$. Sea $(\vec{z},\gamma) \in S$, fijo. Sea $\mathbb{P}$ un procedimiento efectivo que compute a χ_S. Ya vimos en el ejemplo anterior que $\omega^2 \times \Sigma^{*3}$ es Σ -efectivamente enumerable. En forma similar se puede ver que $\omega^n \times \Sigma^{*m}$ lo es. Sea $\mathbb{P}_1$ un procedimiento efectivo que enumere a $\omega^n \times \Sigma^{*m}$. Entonces el siguiente procedimiento enumera a S:

Etapa 1

Realizar $\mathbb{P}_1$ con x de entrada para obtener $(\vec{x},\vec{\alpha}) \in \omega^n \times \Sigma^{*m}$.

Etapa 2

Realizar $\mathbb{P}$ con $(\vec{x},\vec{\alpha})$ de entrada para obtener el valor Booleano e de salida.

Etapa 3

Si $e = 1$ dar como dato de salida $(\vec{x},\vec{\alpha})$. Si $e = 0$ dar como dato de salida $(\vec{z},\gamma)$. $\square$

La resiproca del lema anterior no es cierta. Sin envargo tenemos el siguiente interesante resultado:

Teorema 16 Sea $S \subseteq \omega^n \times \Sigma^{*m}$. Son equivalentes (a) S es Σ-efectivamente computable (b) S y $(\omega^n \times \Sigma^{*m}) - S$ son Σ -efectivamente enumerables Prueba: (a)$\Rightarrow$(b). Por el lema anterior tenemos que S es Σ -efectivamente enumerable. Notese ademas que, dado que S es Σ -efectivamente computable, $(\omega^n \times \Sigma^{*m}) - S$ tambien lo es (por que?). Es decir que aplicando nuevamente el lema anterior tenemos que $(\omega^n \times \Sigma^{*m}) - S$ es Σ-efectivamente enumerable.

(b)$\Rightarrow$(a). Sea $\mathbb{P}_1$ un procedimiento efectivo que enumere a S y sea $\mathbb{P}_2$ un procedimiento efectivo que enumere a $(\omega^n \times \Sigma^{*m}) - S$. Es facil ver que el siguiente procedimiento computa el predicado χ_S:

Etapa 1

Darle a la variable T el valor 0.

Etapa 2

Realizar $\mathbb{P}_1$ con el valor de T como entrada para obtener de salida la upla $(\vec{y},\vec{\beta})$.

Etapa 3

Realizar $\mathbb{P}_2$ con el valor de T como entrada para obtener de salida la upla $(\vec{z},\vec{\gamma})$.

Etapa 4

Si $(\vec{y},\vec{\beta}) = (\vec{x},\vec{\alpha})$, entonces detenerse y dar como dato de salida el valor 1. Si $(\vec{z},\vec{\gamma}) = (\vec{x},\vec{\alpha})$, entonces detenerse y dar como dato de salida el valor 0. Si no suceden ninguna de las dos posibilidades antes mensionadas, aumentar en 1 el valor de la variable T y dirijirse a la Etapa 2. $\square$

Dada una funcion $F : D_F \subseteq \omega^n \times \Sigma^{*m} \to \omega^k \times \Sigma^{*l}$ e $i \in \{1,...,k+l\}$, usaremos F_i para denotar la funcion $p_i^{k,l} \circ F$. Notese que el dominio de cada F_i es igual a D_F.

Teorema 17 Dado $S \subseteq \omega^n \times \Sigma^{*m}$, son equivalentes (1) S es Σ-efectivamente enumerable (2) $S = \varnothing$ o $S = I_F$, para alguna $F : \omega \to \omega^n \times \Sigma^{*m}$ tal que cada F_i es Σ -efectivamente computable. (3) $S = I_F$, para alguna $F : D_F \subseteq \omega^k \times \Sigma^{*l} \to \omega^n \times \Sigma^{*m}$ tal que cada F_i es Σ-efectivamente computable. (4) $S = D_f$, para alguna funcion f la cual es Σ -efectivamente computable. Prueba: (1)$\Rightarrow$(2) y (2)$\Rightarrow$(1) son muy naturales y son dejadas al lector. (2)$\Rightarrow$(3) es trivial.

(3)$\Rightarrow$(4). Para $i = 1,...,n+m$, sea $\mathbb{P}_i$ un procedimiento el cual computa a F_i y sea $\mathbb{P}$ un procedimiento el cual enumere a $\omega \times \omega^k \times \Sigma^{*l}$. El siguiente procedimiento computa la funcion

$f : I_F \to \{1\}$:

Etapa 1

Darle a la variable T el valor 0.

Etapa 2

Hacer correr $\mathbb{P}$ con dato de entrada T y obtener $(t, z_1, ..., z_k, \gamma_1, ..., \gamma_l)$ como dato de salida.

Etapa 3

Para cada $i = 1, ..., n+m$, hacer correr $\mathbb{P}_i$ durante t pasos, con dato de entrada $(z_1, ..., z_k, \gamma_1, ..., \gamma_l)$. Si cada procedimiento $\mathbb{P}_i$ al cabo de los t pasos termino y dio como resultado el valor o_i, entonces comparar $(\vec{x}, \vec{\alpha})$ con $(o_1, ..., o_{n+m})$ y en caso de que sean iguales detenerse y dar como dato de salida el valor 1. En el caso en que no son iguales, aumentar en 1 el valor de la variable T y dirijirse a la Etapa 2. Si algun procedimiento $\mathbb{P}_i$ al cabo de los t pasos no termino, entonces aumentar en 1 el valor de la variable T y dirijirse a la Etapa 2.

(4)$\Rightarrow$(1). Supongamos $S \neq \varnothing$. Sea $(\vec{z}, \vec{\gamma})$ un elemento fijo de S. Sea $\mathbb{P}$ un procedimiento el cual compute a f. Sea $\mathbb{P}_1$ un procedimiento el cual enumere a $\omega \times \omega^n \times \Sigma^{*m}$. Dejamos al lector el dise ño de un procedimiento efectivo el cual enumere D_f. $\square$

0.3. Funciones Σ-recursivas

En esta seccion introducimos el concepto de funcion Σ-recursiva. De la definicion de funcion Σ-recursiva quedara claro que se trata de una familia de funciones que son Σ-efectivamente computables ya que las mismas se obtienen a partir de una familia de funciones muy simples y obviamente Σ-efectivamente computables, usando constructores que preservan la computabilidad efectiva. De hecho, la motivacion en la definicion de funcion Σ-recursiva es lograr una descripcion matematicamente precisa del concepto de funcion Σ-efectivamente computable.

Comensaremos estudiando ciertas funciones Σ-recursivas las cuales tendran un roll fundamental en la teoria

0.3.1. Funciones Σ-recursivas primitivas

Para definir la clase de las funciones Σ-recursivas primitivas necesitaremos dos constructores de funciones a partir de funciones, a saber, la composicion y la recursion primitiva.

Composicion de funciones

Sean

$$\begin{array}{lll} f & : & D_f \subseteq \omega^n \times \Sigma^{*m} \to O, \text{ con } O \in \{\omega, \Sigma^*\} \\ f_i & : & D_{f_i} \subseteq \omega^k \times \Sigma^{*l} \to \omega, \ i = 1, ..., n \\ f_i & : & D_{f_i} \subseteq \omega^k \times \Sigma^{*l} \to \Sigma^*, \ i = n+1, ..., n+m. \end{array}$$

Definamos $f \circ (f_1, ..., f_{n+m}) : D_{f \circ (f_1, ..., f_{n+m})} \subseteq \omega^k \times \Sigma^{*l} \to O$,

de la siguiente manera
$$\begin{array}{rcl} D_{f \circ (f_1, ..., f_{n+m})} & = & \left\{ (\vec{x}, \vec{\alpha}) \in \bigcap_{i=1}^{n+m} D_{f_i} : (f_1(\vec{x}, \vec{\alpha}), ..., f_{n+m}(\vec{x}, \vec{\alpha})) \in D_f \right\} \\ f \circ (f_1, ..., f_{n+m})(\vec{x}, \vec{\alpha}) & = & f(f_1(\vec{x}, \vec{\alpha}), ..., f_{n+m}(\vec{x}, \vec{\alpha})). \end{array}$$

Diremos que la funcion $f \circ (f_1, ..., f_{n+m})$ es obtenida por composicion de $f, f_1, ..., f_{n+m}$. Como es usual, cuando $n+m = 1$, escribiremos $f \circ f_1$ en lugar de $f \circ (f_1)$. Lema 18 Si $f, f_1, ..., f_{n+m}$ son Σ-efectivamente computables, entonces $f \circ (f_1, ..., f_{n+m})$ lo es. Prueba: Sean $\mathbb{P}, \mathbb{P}_1, ..., \mathbb{P}_{n+m}$ procedimientos efectivos los cuales computen las funciones $f, f_1, ..., f_{n+m}$, respectivamente. Usando estos procedimientos es facil definir un procedimiento efectivo el cual compute a $f \circ (f_1, ..., f_{n+m})$. $\square$

Recursion primitiva sobre variable numerica

Caso 1. Sean

$f \ : \ S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$

$g \ : \ \omega \times \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$

con $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ conjuntos no vacios. Definamos $R(f,g) : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$

de la siguiente manera (1) $R(f,g)(0,\vec{x},\vec{\alpha}) = f(\vec{x},\vec{\alpha})$ (2) $R(f,g)(t{+}1,\vec{x},\vec{\alpha}) = g(R(f,g)(t,\vec{x},\vec{\alpha}),t,\vec{x},\vec{\alpha})$. Diremos que $R(f,g)$ es obtenida por recursion primitiva a partir de f y g. Notese que cuando $m = n = 0$, se tiene que $D_f = \{\Diamond\}$ y (1) y (2) se transforman en

(1) $R(f,g)(0) = f(\Diamond)$, (2) $R(f,g)(t+1) = g(R(f,g)(t),t)$. Caso 2. Sean

$f \ : \ S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \Sigma^*$

$g \ : \ \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times \Sigma^* \to \Sigma^*$

con $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ conjuntos no vacios. Definamos $R(f,g) : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \Sigma^*$

de la siguiente manera (1) $R(f,g)(0,\vec{x},\vec{\alpha}) = f(\vec{x},\vec{\alpha})$ (2) $R(f,g)(t{+}1,\vec{x},\vec{\alpha}) = g(t,\vec{x},\vec{\alpha},R(f,g)(t,\vec{x},\vec{\alpha}))$ Diremos que $R(f,g)$ es obtenida por recursion primitiva a partir de f y g.

Lema 19 Si f y g son Σ-efectivamente computables, entonces $R(f,g)$ lo es. Prueba: La prueba es dejada al lector. □

Recursion primitiva sobre variable alfabetica

Caso 1. Sea

$f : S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$

con $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ conjuntos no vacios y sea $\mathcal{G} = \langle \mathcal{G}_a : a \in \Sigma \rangle$ una familia indexada de funciones tal que $\mathcal{G}_a : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times \Sigma^* \to \omega$

para cada $a \in \Sigma$. Definamos $R(f,\mathcal{G}) : S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times \Sigma^* \to \omega$

de la siguiente manera (1) $R(f,\mathcal{G})(\vec{x},\vec{\alpha},\varepsilon) = f(\vec{x},\vec{\alpha})$ (2) $R(f,\mathcal{G})(\vec{x},\vec{\alpha},\alpha a) = \mathcal{G}_a(R(f,\mathcal{G})(\vec{x},\vec{\alpha},\alpha),\vec{x},\vec{\alpha},\alpha)$ Diremos que $R(f,\mathcal{G})$ es obtenida por recursion primitiva a partir de f y $\mathcal{G}$.

Caso 2. Sea

$f : S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \Sigma^*$

con $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ conjuntos no vacios y sea $\mathcal{G} = \langle \mathcal{G}_a : a \in \Sigma \rangle$ una familia indexada de funciones tal que $\mathcal{G}_a : S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times \Sigma^* \times \Sigma^* \to \Sigma^*$

para cada $a \in \Sigma$. Definamos $R(f,\mathcal{G}) : S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times \Sigma^* \to \Sigma^*$

de la siguiente manera (1) $R(f,\mathcal{G})(\vec{x},\vec{\alpha},\varepsilon) = f(\vec{x},\vec{\alpha})$ (2) $R(f,\mathcal{G})(\vec{x},\vec{\alpha},\alpha a) = \mathcal{G}_a(\vec{x},\vec{\alpha},\alpha,R(f,\mathcal{G})(\vec{x},\vec{\alpha},\alpha))$. Diremos que $R(f,\mathcal{G})$ es obtenida por recursion primitiva a partir de f y $\mathcal{G}$.

Lema 20 Si f y cada $\mathcal{G}_a$ son Σ-efectivamente computables, entonces $R(f,\mathcal{G})$ lo es. Prueba: Es dejada al lector con la recomendacion de que haga la prueba para el caso $\Sigma = \{@, \&\}$ □

Definamos los conjuntos $\mathrm{PR}_0^\Sigma \subseteq \mathrm{PR}_1^\Sigma \subseteq \mathrm{PR}_2^\Sigma \subseteq ... \subseteq \mathrm{PR}^\Sigma$ de la siguiente manera

$$\begin{aligned}
\mathrm{PR}_0^\Sigma &= \left\{ Suc, Pred, C_0^{0,0}, C_\varepsilon^{0,0} \right\} \cup \{d_a : a \in \Sigma\} \cup \left\{ p_j^{n,m} : 1 \leq j \leq n+m \right\} \\
\mathrm{PR}_{k+1}^\Sigma &= \mathrm{PR}_k^\Sigma \cup \left\{ f \circ (f_1, ..., f_n) : f, f_1, ..., f_n \in \mathrm{PR}_k^\Sigma \right\} \cup \\
&\qquad \left\{ R(f,\mathcal{G}) : f \text{ y cada } \mathcal{G}_a \text{ pertenecen a } \mathrm{PR}_k^\Sigma \right\} \cup \\
&\qquad \left\{ R(f,g) : f, g \in \mathrm{PR}_k^\Sigma \right\} \\
\mathrm{PR}^\Sigma &= \textstyle\bigcup_{k \geq 0} \mathrm{PR}_k^\Sigma
\end{aligned}$$

Una funcion es llamada Σ-recursiva primitiva (Σ -p.r.) si pertenece a PR^Σ. Para el caso $\Sigma = \varnothing$, notese que

$$\begin{aligned}
\mathrm{PR}_0^\Sigma &= \left\{ Suc, Pred, C_0^{0,0} \right\} \cup \left\{ p_j^{n,0} : 1 \leq j \leq n \right\} \\
\mathrm{PR}_{k+1}^\Sigma &= \mathrm{PR}_k^\Sigma \cup \left\{ f \circ (f_1, ..., f_n) : f, f_1, ..., f_n \in \mathrm{PR}_k^\Sigma \right\} \cup \\
&\qquad \left\{ R(f,g) : f, g \in \mathrm{PR}_k^\Sigma \right\} \\
\mathrm{PR}^\Sigma &= \textstyle\bigcup_{k \geq 0} \mathrm{PR}_k^\Sigma
\end{aligned}$$

Notese ademas que $\mathrm{PR}^{\varnothing} \subseteq \mathrm{PR}^{\Sigma}$, cualquiera sea el alfabeto Σ. Teorema 21 Si $f \in \mathrm{PR}^{\Sigma}$, entonces f es Σ-efectivamente computable. Prueba: Dejamos al lector la prueba por induccion en k de que si $f \in \mathrm{PR}_k^{\Sigma}$, entonces f es Σ-efectivamente computable, la cual sale en forma directa usando los lemas anteriores que garantizan que los constructores de composicion y recursion primitiva preservan la computabilidad efectiva $\Box$

Lema 22 (1) $\varnothing \in \mathrm{PR}^{\varnothing}$. (2) $\lambda xy\,[x+y] \in \mathrm{PR}^{\varnothing}$. (3) $\lambda xy\,[x.y] \in \mathrm{PR}^{\varnothing}$. (4) $\lambda x\,[x!] \in \mathrm{PR}^{\varnothing}$. Prueba: (1) Notese que $\varnothing = Pred \circ C_0^{0,0} \in \mathrm{PR}_1^{\varnothing}$

(2) Notar que

$$\begin{array}{rcl} \lambda xy\,[x+y]\,(0,x_1) & = & x_1 = p_1^{1,0}(x_1) \\ \lambda xy\,[x+y]\,(t+1,x_1) & = & \lambda xy\,[x+y]\,(t,x_1)+1 \\ & = & \left(Suc \circ p_1^{3,0}\right)(\lambda xy\,[x+y]\,(t,x_1),t,x_1) \end{array}$$

lo cual implica que $\lambda xy\,[x+y] = R\left(p_1^{1,0}, Suc \circ p_1^{3,0}\right) \in \mathrm{PR}_2^{\varnothing}$. (3) Primero note que

$$\begin{array}{rcl} C_0^{1,0}(0) & = & C_0^{0,0}(\Diamond) \\ C_0^{1,0}(t+1) & = & C_0^{1,0}(t) \end{array}$$

lo cual implica que $C_0^{1,0} = R\left(C_0^{0,0}, p_1^{2,0}\right) \in \mathrm{PR}_1^{\varnothing}$. Tambien note que $\lambda tx\,[t.x] = R\left(C_0^{1,0}, \lambda xy\,[x+y] \circ \left(p_1^{3,0}\right.\right.$

lo cual por (1) implica que $\lambda tx\,[t.x] \in \mathrm{PR}_3^{\varnothing}$. (4) Note que

$$\begin{array}{rcl} \lambda x\,[x!]\,(0) & = & 1 = C_1^{0,0}(\Diamond) \\ \lambda x\,[x!]\,(t+1) & = & \lambda x\,[x!]\,(t).(t+1), \end{array}$$

lo cual implica que $\lambda x\,[x!] = R\left(C_1^{0,0}, \lambda xy\,[x.y] \circ \left(p_1^{2,0}, Suc \circ p_2^{2,0}\right)\right)$.

Ya que $C_1^{0,0} = Suc \circ C_0^{0,0}$, tenemos que $C_1^{0,0} \in \mathrm{PR}_1^{\varnothing}$. Por (2), tenemos que $\lambda xy\,[x.y] \circ \left(p_1^{2,0}, Suc \circ p_2^{2,0}\right) \in \mathrm{PR}_4^{\varnothing}$,

obteniendo que $\lambda x\,[x!] \in \mathrm{PR}_5^{\varnothing}$. $\Box$

Ahora consideraremos dos funciones las cuales son obtenidas naturalmente por recursion primitiva sobre variable alfabetica.

Lema 23 Supongamos Σ es no vacio. (a) $\lambda\alpha\beta\,[\alpha\beta] \in \mathrm{PR}^{\Sigma}$ (b) $\lambda\alpha\,[|\alpha|] \in \mathrm{PR}^{\Sigma}$ Prueba: (a) Ya que

$$\begin{array}{rcl} \lambda\alpha\beta\,[\alpha\beta]\,(\alpha_1,\varepsilon) & = & \alpha_1 = p_1^{0,1}(\alpha_1) \\ \lambda\alpha\beta\,[\alpha\beta]\,(\alpha_1,\alpha a) & = & d_a(\lambda\alpha\beta\,[\alpha\beta]\,(\alpha_1,\alpha)), a \in \Sigma \end{array}$$

tenemos que $\lambda\alpha\beta\,[\alpha\beta] = R\left(p_1^{0,1},\mathcal{G}\right)$, donde $\mathcal{G}_a = d_a \circ p_3^{0,3}$, para cada $a \in \Sigma$. (b) Ya que

$$\begin{array}{rcl} \lambda\alpha\,[|\alpha|]\,(\varepsilon) & = & 0 = C_0^{0,0}(\Diamond) \\ \lambda\alpha\,[|\alpha|]\,(\alpha a) & = & \lambda\alpha\,[|\alpha|]\,(\alpha)+1 \end{array}$$

tenemos que $\lambda\alpha\,[|\alpha|] = R\left(C_0^{0,0},\mathcal{G}\right)$, donde $\mathcal{G}_a = Suc \circ p_1^{1,1}$, para cada $a \in \Sigma$.. $\Box$ Lema 24 (a) $C_k^{n,m}, C_\alpha^{n,m} \in \mathrm{PR}^{\Sigma}$, para $n,m,k \geq 0$, $\alpha \in \Sigma^*$. (b) $C_k^{n,0} \in \mathrm{PR}^{\varnothing}$, para $n,k \geq 0$. Prueba: (a) Note que $C_{k+1}^{0,0} = Suc \circ C_k^{0,0}$, lo cual implica $C_k^{0,0} \in \mathrm{PR}_k^{\Sigma}$, para $k \geq 0$. Tambien note que $C_{\alpha a}^{0,0} = d_a \circ C_\alpha^{0,0}$, lo cual dice que $C_\alpha^{0,0} \in \mathrm{PR}^{\Sigma}$, para $\alpha \in \Sigma^*$. Para ver que $C_k^{0,1} \in \mathrm{PR}^{\Sigma}$ notar que

$$\begin{array}{rcl} C_k^{0,1}(\varepsilon) & = & k = C_k^{0,0}(\Diamond) \\ C_k^{0,1}(\alpha a) & = & C_k^{0,1}(\alpha) = p_1^{1,1}\left(C_k^{0,1}(\alpha),\alpha\right) \end{array}$$

lo cual implica que $C_k^{0,1} = R\left(C_k^{0,0},\mathcal{G}\right)$, con $\mathcal{G}_a = p_1^{1,1}$, $a \in \Sigma$. En forma similar podemos ver que $C_k^{1,0}, C_\alpha^{1,0}, C_\alpha^{0,1} \in \mathrm{PR}^{\Sigma}$. Supongamos ahora que $m > 0$. Entonces $\begin{array}{rcl} C_k^{n,m} & = & C_k^{0,1} \circ p_{n+1}^{n,m} \\ C_\alpha^{n,m} & = & C_\alpha^{0,1} \circ p_{n+1}^{n,m} \end{array}$

de lo cual obtenemos que $C_k^{n,m}, C_\alpha^{n,m} \in \mathrm{PR}^{\Sigma}$. El caso $n > 0$ es similar (b) Use argumentos similares a los usados en la prueba de (a). $\Box$

Definamos $0^0 = 1$. O sea que $D_{\lambda xy[x^y]} = \omega \times \omega$. Tambien ya que $\alpha^0 = \varepsilon$, tenemos que $D_{\lambda x\alpha[\alpha^x]} = \omega \times \Sigma^*$.

Lema 25 (a) $\lambda xy\,[x^y] \in \mathrm{PR}^{\varnothing}$. (b) $\lambda t\alpha\,[\alpha^t] \in \mathrm{PR}^{\Sigma}$. Prueba: (a) Note que

$\lambda tx\left[x^t\right] = R\left(C_1^{1,0},\lambda xy\,[x.y] \circ \left(p_1^{3,0},p_3^{3,0}\right)\right) \in \mathrm{PR}^{\varnothing}$.

O sea que $\lambda xy\,[x^y] = \lambda tx\,[x^t] \circ \left(p_2^{2,0},p_1^{2,0}\right) \in \mathrm{PR}^{\varnothing}$. (b) Note que

$\lambda t\alpha\left[\alpha^t\right] = R\left(C_\varepsilon^{0,1},\lambda\alpha\beta\,[\alpha\beta] \circ \left(p_3^{1,2},p_2^{1,2}\right)\right) \in \mathrm{PR}^{\Sigma}$.

$\Box$ Ahora probaremos que si Σ es no vacio, entonces las biyeciones naturales entre Σ^* y ω, dadas en el Lema 6, son Σ-p.r..

Lema 26 Si $<$ es un orden total estricto sobre un alfabeto no vacio Σ, entonces $s^<$, $\#^<$ y $*^<$ pertenecen a PR^{Σ} Prueba: Supongamos $\Sigma = \{a_1,...,a_k\}$ y $<$ dado por $a_1 < ... < a_k$. Ya que

$$\begin{array}{rcl} s^<(\varepsilon) & = & a_1 \\ s^<(\alpha a_i) & = & \alpha a_{i+1}, \text{ para } i < k \\ s^<(\alpha a_k) & = & s^<(\alpha)a_1 \end{array}$$

tenemos que $s^< = R\left(C_{a_1}^{0,0},\mathcal{G}\right)$, donde $\mathcal{G}_{a_i} = d_{a_{i+1}} \circ p_1^{0,2}$, para $i = 1,...,k-1$ y $\mathcal{G}_{a_k} = d_{a_1} \circ p_2^{0,2}$. O sea que $s^< \in \mathrm{PR}^{\Sigma}$. Ya que $\begin{array}{rcl} *^<(0) & = & \varepsilon \\ *^<(t+1) & = & s^<(*^<(t)) \end{array}$

podemos ver que $*^< \in \mathrm{PR}^{\Sigma}$. Ya que $\begin{array}{rcl} \#^<(\varepsilon) & = & 0 \\ \#^<(\alpha a_i) & = & \#^<(\alpha).k+i, \text{ para } i = 1,...,k, \end{array}$

tenemos que $\#^< = R\left(C_0^{0,0},\mathcal{G}\right)$, donde $\mathcal{G}_{a_i} = \lambda xy\,[x+y] \circ \left(\lambda xy\,[x.y] \circ \left(p_1^{1,1},C_k^{1,1}\right),C_i^{1,1}\right)$, para $i = 1,...,k$.

O sea que $\#^< \in \mathrm{PR}^{\Sigma}$. $\Box$ Dados $x,y \in \omega$, definamos

$x\dot{-}y = \max(x-y,0)$.

Lema 27 (a) $\lambda xy\,[x\dot{-}y] \in \mathrm{PR}^{\varnothing}$. (b) $\lambda xy\,[\max(x,y)] \in \mathrm{PR}^{\varnothing}$. (c) $\lambda xy\,[x=y] \in \mathrm{PR}^{\varnothing}$. (d) $\lambda xy\,[x \leq y] \in \mathrm{PR}^{\varnothing}$. (e) Si Σ es no vacio, entonces $\lambda\alpha\beta\,[\alpha=\beta] \in \mathrm{PR}^{\Sigma}$ Prueba: (a) Primero notar que $\lambda x\,[x\dot{-}1] = R\left(C_0^{0,0},p_2^{2,0}\right) \in \mathrm{PR}^{\varnothing}$. Tambien note que

$\lambda tx\,[x\dot{-}t] = R\left(p_1^{1,0},\lambda x\,[x\dot{-}1] \circ p_1^{3,0}\right) \in \mathrm{PR}^{\varnothing}$.

O sea que $\lambda xy\left[x\dot{-}y\right] = \lambda tx\left[x\dot{-}t\right] \circ \left(p_2^{2,0}, p_1^{2,0}\right) \in \mathrm{PR}^{\varnothing}$. (b) Note que $\lambda xy\left[\max(x,y)\right] = \lambda xy\left[(x+(y\dot{-}x)\right]$.

(c) Note que $\lambda xy\left[x=y\right] = \lambda xy\left[1\dot{-}((x\dot{-}y)+(y\dot{-}x))\right]$.

(d) Note que $\lambda xy\left[x\leq y\right] = \lambda xy\left[1\dot{-}(x\dot{-}y)\right]$.

(e) Sea $<$ un orden total estricto sobre Σ. Ya que

$\alpha=\beta$ sii $\#^{<}(\alpha)=\#^{<}(\beta)$

tenemos que $\lambda\alpha\beta\left[\alpha=\beta\right] = \lambda xy\left[x=y\right]\circ\left(\#^{<}\circ p_1^{0,2}, \#^{<}\circ p_2^{0,2}\right)$.

O sea que podemos aplicar (c) y Lema 28 implica que χ_S es Σ-p.r.. □ El siguiente lema caracteriza cuando un conjunto rectangular es Σ -p.r..

Lema 31 Supongamos $S_1,...,S_n\subseteq\omega$, $L_1,...,L_m\subseteq\Sigma^*$ son conjuntos no vacios. Entonces $S_1\times...\times S_n\times L_1\times...\times L_m$ es Σ -p.r. sii $S_1,...,S_n,L_1,...,L_m$ son Σ-p.r. Prueba: ($\Rightarrow$) Veremos por ejemplo que L_1 es Σ-p.r.. Sea $(z_1,...,z_n,\zeta_1,...,\zeta_m)$ un elemento fijo de $S_1\times...\times S_n\times L_1\times...\times L_m$. Note que

$\alpha\in L_1$ sii $(z_1,...,z_n,\alpha,\zeta_2,...,\zeta_m)\in S_1\times...\times S_n\times L_1\times...\times L_m$,

lo cual implica que $\chi_{L_1}=\chi_{S_1\times...\times S_n\times L_1\times...\times L_m}\circ\left(C_{z_1}^{0,1},...,C_{z_n}^{0,1},p_1^{0,1},C_{\zeta_2}^{0,1},...,C_{\zeta_m}^{0,1}\right)$.

($\Leftarrow$) Note que $\chi_{S_1\times...\times S_n\times L_1\times...\times L_m}$ es el predicado $(\chi_{S_1}\circ p_1^{n,m}\wedge...\wedge\chi_{S_n}\circ p_n^{n,m}\wedge\chi_{L_1}\circ p_{n+1}^{n,m}\wedge...\wedge\chi_{L_m}\circ$

□ Dada una funcion f y un conjunto $S\subseteq D_f$, usaremos $f\mid_S$ para denotar la restriccion de f al conjunto S, i.e. $f\mid_S=f\cap(S\times I_f)$.

Lema 32 Supongamos $f:D_f\subseteq\omega^n\times\Sigma^{*m}\to O$ es Σ-p.r., donde $O\in\{\omega,\Sigma^*\}$. Si $S\subseteq D_f$ es Σ-p.r., entonces $f\mid_S$ es Σ-p.r.. Prueba: Supongamos $O=\Sigma^*$. Entonces

$f\mid_S=\lambda x\alpha\left[\alpha^x\right]\circ(Suc\circ Pred\circ\chi_S,f)$

es Σ-p.r.. El caso $O=\omega$ es similar usando $\lambda xy\left[x^y\right]$ en lugar de $\lambda x\alpha\left[\alpha^x\right]$. □ Usando el lema anterior en combinacion con el Lema 28 podemos ver que muchos predicados usuales son Σ-p.r.. Por ejemplo sea

$P=\lambda x\alpha\beta\gamma\left[x=|\gamma|\wedge\alpha=\gamma^{Pred(|\beta|)}\right]$.

Notese que $D_P=\omega\times\Sigma^*\times(\Sigma^*-\{\varepsilon\})\times\Sigma^*$

es Σ-p.r. ya que $\chi_{D_P}=\neg\lambda\alpha\beta\left[\alpha=\beta\right]\circ\left(p_3^{1,3},C_\varepsilon^{1,3}\right)$.

Tambien note que los predicados $\begin{array}{l}\lambda x\alpha\beta\gamma\left[x=|\gamma|\right]\\ \lambda x\alpha\beta\gamma\left[\alpha=\gamma^{Pred(|\beta|)}\right]\end{array}$

son Σ-p.r. ya que pueden obtenerse componiendo funciones Σ -p.r.. O sea que P es Σ-p.r. ya que $P=\left(\lambda x\alpha\beta\gamma\left[x=|\gamma|\right]\mid_{D_P}\wedge\lambda x\alpha\beta\gamma\left[\alpha=\gamma^{Pred(|\beta|)}\right]\right)$.

Lema 33 Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ es Σ-p.r., entonces existe una funcion Σ -p.r. $\bar{f} : \omega^n \times \Sigma^{*m} \rightarrow O$, tal que $f = \bar{f} \mid_{D_f}$. Prueba: Es facil ver por induccion en k que el enunciado se cumple para cada $f \in \mathrm{PR}_k^{\Sigma}$ $\Box$

Proposición 34 Un conjunto S es Σ-p.r. sii S es el dominio de una funcion Σ-p.r.. Prueba: $(\Rightarrow)$ Note que $S = D_{Pred \circ \chi_S}$.

$(\Leftarrow)$ Probaremos por induccion en k que D_F es Σ -p.r., para cada $F \in \mathrm{PR}_k^{\Sigma}$. El caso $k = 0$ es facil. Supongamos el resultado vale para un k fijo y supongamos $F \in \mathrm{PR}_{k+1}^{\Sigma}$. Veremos entonces que D_F es Σ-p.r.. Hay varios casos. Consideremos primero el caso en que $F = R(f, g)$, donde

$$f : S_1 \times ... \times S_n \times L_1 \times ... \times L_m \rightarrow \Sigma^*$$
$$g : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times \Sigma^* \rightarrow \Sigma^*,$$

con $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ conjuntos no vacios y $f, g \in \mathrm{PR}_k^{\Sigma}$. Notese que por definicion de $R(f, g)$, tenemos que $D_F = \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m$.

Por hipotesis inductiva tenemos que $D_f = S_1 \times ... \times S_n \times L_1 \times ... \times L_m$ es Σ-p.r., lo cual por el Lema 31 nos dice que los conjuntos $S_1, ..., S_n$, $L_1, ..., L_m$ son Σ-p.r.. Ya que ω es Σ-p.r., el Lema 31 nos dice que D_F es Σ-p.r.. Los otros casos de recursion primitiva son dejados al lector.

Supongamos ahora que $F = g \circ (g_1, ..., g_{n+m})$, donde

$$g : D_g \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$$
$$g_i : D_{g_i} \subseteq \omega^k \times \Sigma^{*l} \rightarrow \omega,\ i = 1, ..., n$$
$$g_i : D_{g_i} \subseteq \omega^k \times \Sigma^{*l} \rightarrow \Sigma^*, i = n + 1, ..., n + m$$

estan en PR_k^{Σ}. Por Lema 33, hay funciones Σ-p.r. $\bar{g}_1, ..., \bar{g}_{n+m}$ las cuales son Σ-totales y cumplen $g_i = \bar{g}_i \mid_{D_{g_i}}$, para $i = 1, ..., n + m$.

Por hipotesis inductiva los conjuntos D_g, D_{g_i}, $i = 1, ..., n + m$, son Σ-p.r. y por lo tanto $S = \bigcap_{i=1}^{n+m} D_{g_i}$

lo es. Notese que $\chi_{D_F} = (\chi_{D_g} \circ (\bar{g}_1, ..., \bar{g}_{n+m}) \wedge \chi_S)$

lo cual nos dice que D_F es Σ-p.r.. $\Box$

Lema de division por casos

Una observacion interesante es que si $f_i : D_{f_i} \rightarrow O$, $i = 1, ..., k$, son funciones tales que $D_{f_i} \cap D_{f_j} = \varnothing$ para $i \neq j$, entonces $f_1 \cup ... \cup f_k$ es la funcion

$$\begin{array}{rcl} D_{f_1} \cup ... \cup D_{f_k} & \rightarrow & O \\ e & \rightarrow & \left\{ \begin{array}{ll} f_1(e) & \text{si } e \in D_{f_1} \\ \vdots & \vdots \\ f_k(e) & \text{si } e \in D_{f_k} \end{array} \right. \end{array}$$

Lema 35 Supongamos $f_i : D_{f_i} \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$, $i = 1, ..., k$, son funciones Σ-p.r. tales que $D_{f_i} \cap D_{f_j} = \varnothing$ para $i \neq j$. Entonces $f_1 \cup ... \cup f_k$ es Σ-p.r.. Prueba: Supongamos $O = \Sigma^*$ y $k = 2$. Sean

$\bar{f}_i : \omega^n \times \Sigma^{*m} \rightarrow \Sigma^*, i = 1, 2,$

funciones Σ-p.r. tales que $\bar{f}_i \mid_{D_{f_i}} = f_i$, $i = 1, 2$ (Lema 33). Por Lema 34 los conjuntos D_{f_1} y D_{f_2} son Σ-p.r. y por lo tanto lo es $D_{f_1} \cup D_{f_2}$. Ya que $f_1 \cup f_2 = \Big(\lambda\alpha\beta\left[\alpha\beta\right] \circ (\lambda x\alpha\left[\alpha^x\right] \circ (\chi_{D_{f_1}}, \bar{f}_1), \lambda x\alpha\left[\alpha^x\right] \circ (\chi$

tenemos que $f_1 \cup f_2$ es Σ-p.r.. El caso $k > 2$ puede probarse por induccion ya que

$f_1 \cup ... \cup f_k = (f_1 \cup ... \cup f_{k-1}) \cup f_k$.

$\Box$

Corolario 36 Supongamos f es una funcion Σ-mixta cuyo dominio es finito. Entonces f es Σ-p.r.. Prueba: Supongamos $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$, con $D_f = \{e_1, ..., e_k\}$. Por el Corolario 30, cada $\{e_i\}$ es Σ-p.r. por lo cual el Lema 32 nos dice que $C^{n,m}_{f(e_i)} \mid_{\{e_1\}}$ es Σ-p.r.. O sea que

$f = C^{n,m}_{f(e_1)} \mid_{\{e_1\}} \cup ... \cup C^{n,m}_{f(e_k)} \mid_{\{e_k\}}$

es Σ-p.r.. $\Box$ Recordemos que dados $i \in \omega$ y $\alpha \in \Sigma^*$, definimos

$$[\alpha]_i = \begin{cases} i\text{-esimo elemento de } \alpha & \text{si } 1 \leq i \leq |\alpha| \\ \varepsilon & \text{caso contrario} \end{cases}$$

Lema 37 $\lambda i\alpha\left[[\alpha]_i\right]$ es Σ-p.r.. Prueba: Note que

$$\begin{array}{rcl} [\varepsilon]_i & = & \varepsilon \\ [\alpha a]_i & = & \begin{cases} [\alpha]_i & \text{si } i \neq |\alpha|+1 \\ a & \text{si } i = |\alpha|+1 \end{cases} \end{array}$$

lo cual dice que $\lambda i\alpha\left[[\alpha]_i\right] = R\left(C_\varepsilon^{1,0}, \mathcal{G}\right)$, donde $\mathcal{G}_a : \omega\times\Sigma^*\times\Sigma^* \to \Sigma^*$ es dada por $\mathcal{G}_a(i,\alpha,\zeta) = \begin{cases} \zeta & \text{si } i \neq |\alpha|+1 \\ a & \text{si } i = |\alpha|+1 \end{cases}$

O sea que solo resta probar que cada $\mathcal{G}_a$ es Σ-p.r.. Primero note que los conjuntos $\begin{array}{rcl} S_1 & = & \{(i,\alpha,\zeta) \in \omega \times \\ S_2 & = & \{(i,\alpha,\zeta) \in \omega \times \end{array}$

son Σ-p.r. ya que $\begin{array}{rcl} \chi_{S_1} & = & \lambda xy\left[x \neq y\right] \circ \left(p_1^{1,2}, Suc \circ \lambda\alpha\left[|\alpha|\right] \circ p_2^{1,2}\right) \\ \chi_{S_2} & = & \lambda xy\left[x = y\right] \circ \left(p_1^{1,2}, Suc \circ \lambda\alpha\left[|\alpha|\right] \circ p_2^{1,2}\right). \end{array}$

Ya que $\mathcal{G}_a = p_3^{1,2}|_{S_1} \cup C_a^{1,2}|_{S_2}$,

el Lema 35 nos dice que $\mathcal{G}_a$ es Σ-p.r., para cada $a \in \Sigma$. $\Box$

Sumatoria, productoria y concatenatoria

Sea $f : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$, donde $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ son no vacios. Para $x, y \in \omega$ y $(\vec{x}, \vec{\alpha}) \in S_1 \times ... \times S_n \times L_1 \times ... \times L_m$, definamos

$$\begin{array}{rcl} \sum\limits_{t=x}^{t=y} f(t,\vec{x},\vec{\alpha}) & = & \begin{cases} 0 & \text{si } x > y \\ f(x,\vec{x},\vec{\alpha}) + f(x+1,\vec{x},\vec{\alpha}) + ... + f(y,\vec{x},\vec{\alpha}) & \text{si } x \leq y \end{cases} \\ \prod\limits_{t=x}^{t=y} f(t,\vec{x},\vec{\alpha}) & = & \begin{cases} 1 & \text{si } x > y \\ f(x,\vec{x},\vec{\alpha}).f(x+1,\vec{x},\vec{\alpha})....f(y,\vec{x},\vec{\alpha}) & \text{si } x \leq y \end{cases} \end{array}$$

En forma similar, cuando $I_f \subseteq \Sigma^*$, definamos $\underset{t=x}{\overset{t=y}{\subset}} f(t,\vec{x},\vec{\alpha}) = \begin{cases} \varepsilon & \text{si} \\ f(x,\vec{x},\vec{\alpha})f(x+1,\vec{x},\vec{\alpha})....f(y,\vec{x},\vec{\alpha}) & \text{si} \end{cases}$

Note que, en virtud de la definicion anterior, el dominio de las funciones $\lambda xy\vec{x}\vec{\alpha}\left[\sum\limits_{t=x}^{t=y} f(t,\vec{x},\vec{\alpha})\right]$ λxy

es $\omega \times \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m$.

Lema 38 Sean $n, m \geq 0$. (a) Si $f : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$ es Σ-p.r., con $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ no vacios, entonces lo son las funciones $\lambda xy\vec{x}\vec{\alpha}\left[\sum_{t=x}^{t=y} f(t,\vec{x},\vec{\alpha})\right]$ y $\lambda xy\vec{x}\vec{\alpha}\left[\prod_{t=x}^{t=y} f(t,\vec{x},\vec{\alpha})\right]$. (b) Si $f : \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \Sigma^*$ es Σ-p.r., con $S_1, ..., S_n \subseteq \omega$ y $L_1, ..., L_m \subseteq \Sigma^*$ no vacios, entonces lo es la funcion $\lambda xy\vec{x}\vec{\alpha}\left[\subset_{t=x}^{t=y} f(t,\vec{x},\vec{\alpha})\right]$

Prueba: (a) Sea $G = \lambda tx\vec{x}\vec{\alpha}\left[\sum_{i=x}^{i=t} f(i,\vec{x},\vec{\alpha})\right]$. Ya que

$$\lambda xy\vec{x}\vec{\alpha}\left[\sum_{i=x}^{i=y} f(i,\vec{x},\vec{\alpha})\right] = G \circ \left(p_2^{n+2,m}, p_1^{n+2,m}, p_3^{n+2,m}, ..., p_{n+m+2}^{n+2,m}\right)$$

solo tenemos que probar que G es Σ-p.r.. Primero note que $\begin{array}{rcl} G(0,x,\vec{x},\vec{\alpha}) & = & \begin{cases} 0 & \text{si } x > \\ f(0,\vec{x},\vec{\alpha}) & \text{si } x = \end{cases} \\ G(t+1,x,\vec{x},\vec{\alpha}) & = & \begin{cases} 0 \\ G(t,x,\vec{x},\vec{\alpha}) + f(t+ \end{cases} \end{array}$

Sean $\begin{array}{rcl} D_1 & = & \{(x,\vec{x},\vec{\alpha}) \in \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m : x > 0\} \\ D_2 & = & \{(x,\vec{x},\vec{\alpha}) \in \omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m : x = 0\} \\ H_1 & = & \{(z,t,x,\vec{x},\vec{\alpha}) \in \omega^3 \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m : x > t+1\} \\ H_2 & = & \{(z,t,x,\vec{x},\vec{\alpha}) \in \omega^3 \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m : x \leq t+1\}. \end{array}$

Es facil de chequear que estos conjuntos son Σ-p.r.. Veamos que por ejemplo H_1 lo es. Es decir debemos ver que χ_{H_1} es Σ -p.r.. Ya que f es Σ-p.r. tenemos que $D_f = \omega\times S_1\times...\times S_n\times L_1\times...\times L_m$ es Σ -p.r., lo cual por el Lema 31 nos dice que los conjuntos $S_1, ..., S_n$, $L_1, ..., L_m$ son Σ-p.r..

Ya que ω es Σ-p.r., el Lema 31 nos dice que $R = \omega^3 \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m$ es Σ-p.r.. Notese que $\chi_{H_1} = (\chi_R \wedge \lambda ztx\vec{x}\vec{\alpha}\,[x > t+1])$ por cual χ_{H_1} es Σ -p.r. ya que es la conjuncion de dos predicados Σ-p.r. Ademas note que $G = R(h,g)$, donde

$$\begin{array}{rcl} h & = & C_0^{n+1,m}\mid_{D_1} \cup \lambda x\vec{x}\vec{\alpha}\,[f(0,\vec{x},\vec{\alpha})]\mid_{D_2} \\ g & = & C_0^{n+3,m}\mid_{H_1} \cup \lambda ztx\vec{x}\vec{\alpha}\,[z + f(t+1,\vec{x},\vec{\alpha})])\mid_{H_2} \end{array}$$

O sea que los Lemas 35 y 32 garantizan que G es Σ-p.r.. □ Cuantificacion acotada de predicados con dominio rectangular.

Lema 39 Sean $n,m \geq 0$. (a) Sea $P : S \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \omega$ un predicado Σ-p.r. y supongamos $\bar{S} \subseteq S$ es Σ-p.r.. Entonces $\lambda x\vec{x}\vec{\alpha}\left[(\forall t \in \bar{S})_{t\leq x}\ P(t,\vec{x},\vec{\alpha})\right]$ y $\lambda x\vec{x}\vec{\alpha}\left[(\exists t \in \bar{S})_{t\leq x}\ P(t,\vec{x},\vec{\alpha})\right]$ son predicados Σ-p.r.. (Note que el dominio de estos predicados es $\omega \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m$) (b) Sea $P : S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times L \to \omega$ un predicado Σ-p.r. y supongamos $\bar{L} \subseteq L$ es Σ-p.r.. Entonces $\lambda x\vec{x}\vec{\alpha}\left[(\forall \alpha \in \bar{L})_{|\alpha|\leq x}\ P(\vec{x},\vec{\alpha},\alpha)\right]$ y $\lambda x\vec{x}\vec{\alpha}\left[(\exists \alpha \in \bar{L})_{|\alpha|\leq x}\ P(\vec{x},\vec{\alpha},\alpha)\right]$ son predicados Σ-p.r.. Prueba: (a) Sea

$\bar{P} = P\mid_{\bar{S}\times S_1\times...\times S_n\times L_1\times...\times L_m} \cup C_1^{1+n,m}\mid_{(\omega-\bar{S})\times S_1\times...\times S_n\times L_1\times...\times L_m}$

Notese que $\bar{P}$ es Σ-p.r.. Ya que

$$\begin{array}{rcl} \lambda x\vec{x}\vec{\alpha}\left[(\forall t \in \bar{S})_{t\leq x} P(t,\vec{x},\vec{\alpha})\right] & = & \lambda x\vec{x}\vec{\alpha}\left[\prod\limits_{t=0}^{t=x} \bar{P}(t,\vec{x},\vec{\alpha})\right] \\ & = & \lambda xy\vec{x}\vec{\alpha}\left[\prod\limits_{t=x}^{t=y} \bar{P}(t,\vec{x},\vec{\alpha})\right] \circ \left(C_0^{1+n,m}, p_1^{1+n}\right. \end{array}$$

el Lema 38 implica que $\lambda x\vec{x}\vec{\alpha}\left[(\forall t \in \bar{S})_{t\leq x}\ P(t,\vec{x},\vec{\alpha})\right]$ es Σ-p.r.. Finalmente note que

$\lambda x\vec{x}\vec{\alpha}\left[(\exists t \in \bar{S})_{t\leq x}\ P(t,\vec{x},\vec{\alpha})\right] = \neg\lambda x\vec{x}\vec{\alpha}\left[(\forall t \in \bar{S})_{t\leq x}\ \neg P(t,\vec{x},\vec{\alpha})\right]$

es Σ-p.r.. (b) Sea $<$ un orden total estricto sobre Σ. Sea k el cardinal de Σ. Ya que

$|\alpha| \leq x$ sii $\#^<(\alpha) \leq \sum\limits_{\iota=1}^{i=x} k^i$,

(ejercicio) tenemos que $\lambda x\vec{x}\vec{\alpha}\left[(\forall \alpha \in \bar{L})_{|\alpha|\leq x} P(\vec{x},\vec{\alpha},\alpha)\right] = \lambda x\vec{x}\vec{\alpha}\left[(\forall t \in \#^<(\bar{L}))_{t\leq\sum_{\iota=1}^{i=x} k^i} P(\vec{x},\vec{\alpha},*^<(t))\right]$

Sea $H = \lambda t\vec{x}\vec{\alpha}\,[P(\vec{x},\vec{\alpha},*^<(t))]$. Notese que H es Σ-p.r. y $D_H = \#^<(L) \times S_1 \times ... \times S_n \times L_1 \times ... \times L_m$

Ademas note que $\#^<(\bar{L})$ es Σ-p.r. (ejercicio), lo cual por (a) implica que $Q = \lambda x\vec{x}\vec{\alpha}\left[(\forall t \in \#^<(\bar{L}))_{t\leq x} H(t\right.$

es Σ-p.r.. O sea que $\lambda x\vec{x}\vec{\alpha}\left[(\forall \alpha \in \bar{L})_{|\alpha|\leq x}\ P(\vec{x},\vec{\alpha},\alpha)\right] = Q \circ \left(\lambda x\vec{x}\vec{\alpha}\left[\sum\limits_{\iota=1}^{i=x} k^i\right], p_1^{1+n,m}, ..., p_{1+n+m}^{1+n,m}\right)$

es Σ-p.r.. □ Algunos ejemplos en los cuales cuantificacion acotada se aplica naturalmente son

Lema 40 (a) El predicado $\lambda xy\,[x$ divide $y]$ es $\varnothing$-p.r.. (b) El predicado $\lambda x\,[x$ es primo$]$ es $\varnothing$-p.r.. (c) El predicado $\lambda\alpha\beta\,[\alpha$ inicial $\beta]$ es Σ-p.r.. Prueba: (a) Si tomamos $P = \lambda tx_1x_2\,[x_2 = t.x_1] \in \mathrm{PR}^{\varnothing}$, tenemos que

$$\begin{array}{rcl} \lambda x_1x_2\,[x_1 \text{ divide } x_2] & = & \lambda x_1x_2\,[(\exists t \in \omega)_{t\leq x_2}\ P(t,x_1,x_2)] \\ & = & \lambda xx_1x_2\,[(\exists t \in \omega)_{t\leq x}\ P(t,x_1,x_2)] \circ \left(p_2^{2,0}, p_1^{2,0}, p_2^{2,0}\right) \end{array}$$

por lo que podemos aplicar el lema anterior. (b) Ya que

x es primo sii $x > 1 \wedge ((\forall t \in \omega)_{t\leq x}\ t = 1 \vee t = x \vee \neg(t \text{ divide } x))$

podemos usar un argumento similar al de la prueba de (a). (c) es dejado al lector. □

0.3.2. Minimizacion y funciones Σ-recursivas

Para obtener la clase de las funciones Σ-recursivas debemos agregar una nueva regla a las ya definidas de composicion y recursion primitiva.

Minimizacion de variable numerica

Dado un predicado $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \to \omega$ definimos

$M(P) : D_{M(P)} \subseteq \omega^n \times \Sigma^{*m} \to \omega$

de la siguiente manera
$$\begin{array}{rcl} D_{M(P)} & = & \{(\vec{x},\vec{\alpha}) \in \omega^n \times \Sigma^{*m} : (\exists t \in \omega)\ P(t,\vec{x},\vec{\alpha})\} \\ M(P)(\vec{x},\vec{\alpha}) & = & \text{mín}_t\, P(t,\vec{x},\vec{\alpha}), \text{ para cada } (\vec{x},\vec{\alpha}) \in D_{M(P)} \end{array}$$

Es decir $M(P)$ es exactamente la funcion $\lambda\vec{x}\vec{\alpha}\,[\text{mín}_t\, P(t,\vec{x},\vec{\alpha})]$ ya que la expresion $\text{mín}_t\, P(t,\vec{x},\vec{\alpha})$ esta definida exactamente para aquellas $(n+m)$-uplas $(\vec{x},\vec{\alpha})$ para las cuales hay al menos un t tal que se da $P(t,\vec{x},\vec{\alpha}) = 1$. Diremos que $M(P)$ se obtiene por minimizacion de variable numerica a partir de P.

Lema 41 Si $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \to \omega$ es un predicado Σ-efectivamente computable y D_P es Σ-efectivamente computable, entonces la funcion $M(P)$ es Σ-efectivamente computable.
Prueba: Ejercicio $\Box$

Lamentablemente si quitamos la hipotesis en el lema anterior de que P sea Σ-total, el lema resulta falso. Mas adelante veremos un contraejemplo. Por el momento el lector puede convencerse de que aun teniendo un procedimiento efectivo que compute a un predicado $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \to \omega$ no es claro como construir un procedimiento efectivo que compute a $M(P)$.

Con este nuevo constructor de funciones estamos en condiciones de definir la clase de las funciones Σ-recursivas. Definamos los conjuntos $\mathrm{R}_0^\Sigma \subseteq \mathrm{R}_1^\Sigma \subseteq \mathrm{R}_2^\Sigma \subseteq ... \subseteq \mathrm{R}^\Sigma$ de la siguiente manera

$$\begin{array}{rcl} \mathrm{R}_0^\Sigma & = & \mathrm{PR}_0^\Sigma \\ \mathrm{R}_{k+1}^\Sigma & = & \mathrm{R}_k^\Sigma \cup \left\{f \circ (f_1,...,f_n) : f, f_1,...,f_n \in \mathrm{R}_k^\Sigma\right\} \cup \\ & & \left\{R(f,\mathcal{G}) : f \text{ y cada } \mathcal{G}_a \text{ pertenecen a } \mathrm{R}_k^\Sigma\right\} \cup \\ & & \left\{R(f,g) : f,g \in \mathrm{R}_k^\Sigma\right\} \cup \\ & & \left\{M(P) : P \text{ es } \Sigma\text{-total y } P \in \mathrm{R}_k^\Sigma\right\} \\ \mathrm{R}^\Sigma & = & \bigcup_{k\geq 0} \mathrm{R}_k^\Sigma \end{array}$$

Una funcion f es llamada Σ-recursiva si pertenece a R^Σ. Cabe destacar que aunque $M(P)$ fue definido para predicados no necesariamente Σ-totales, en la definicion de los conjuntos R_k^Σ, nos restringimos al caso en que P es Σ-total. Para el caso $\Sigma = \varnothing$, notese que

$$\begin{array}{rcl} \mathrm{R}_0^\Sigma & = & \mathrm{PR}_0^\Sigma = \left\{Suc, Pred, C_0^{0,0}\right\} \cup \left\{p_j^{n,0} : 1 \leq j \leq n\right\} \\ \mathrm{R}_{k+1}^\Sigma & = & \mathrm{R}_k^\Sigma \cup \left\{f \circ (f_1,...,f_n) : f, f_1,...,f_n \in \mathrm{R}_k^\Sigma\right\} \cup \\ & & \left\{R(f,g) : f,g \in \mathrm{R}_k^\Sigma\right\} \cup \\ & & \left\{M(P) : P \text{ es total y } P \in \mathrm{R}_k^\Sigma\right\} \\ \mathrm{R}^\Sigma & = & \bigcup_{k\geq 0} \mathrm{R}_k^\Sigma \end{array}$$

Notese que $\mathrm{R}^\varnothing \subseteq \mathrm{R}^\Sigma$. Cabe tambien notar que $\mathrm{PR}^\Sigma \subseteq \mathrm{R}^\Sigma$ y $\mathrm{PR}_k^\Sigma \subseteq \mathrm{R}_k^\Sigma$, para cada $k \in \omega$.

Teorema 42 Si $f \in \mathrm{R}^\Sigma$, entonces f es Σ-efectivamente computable. Prueba: Dejamos al lector la prueba por induccion en k de que si $f \in \mathrm{R}_k^\Sigma$, entonces f es Σ-efectivamente computable. $\Box$

Aunque no siempre que $P \in \mathrm{R}^\Sigma$, tendremos que $M(P) \in \mathrm{R}^\Sigma$, el siguiente lema nos garantiza que este es el caso cuando $P \in \mathrm{PR}^\Sigma$ y ademas da condiciones sobre P para que $M(P)$ sea Σ-p.r..

Lema 43 Sean $n,m \geq 0$. Sea $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \to \omega$ un predicado Σ -p.r.. Entonces (a) $M(P)$ es Σ-recursiva. (b) Si hay una funcion Σ-p.r. $f : \omega^n \times \Sigma^{*m} \to \omega$ tal que $M(P)(\vec{x},\vec{\alpha}) = \min\limits_t P(t,\vec{x},\vec{\alpha}) \leq f(\vec{x},\vec{\alpha})$, para cada $(\vec{x},\vec{\alpha}) \in D_{M(P)}$, entonces $M(P)$ es Σ-p.r..
Prueba: (a) Sea $\bar{P} = P\mid_{D_P} \cup C_0^{n+1,m}\mid_{(\omega^{n+1}\times\Sigma^{*m})-D_P}$. Dejamos al lector verificar cuidadosamente

que $M(P) = M(\bar{P})$. Veremos entonces que $M(\bar{P})$ es Σ-recursiva. Note que $\bar{P}$ es Σ-p.r. (por que?). Sea k tal que $\bar{P} \in \mathrm{PR}_k^\Sigma$. Ya que $\bar{P}$ es Σ-total y $\bar{P} \in \mathrm{PR}_k^\Sigma \subseteq \mathrm{R}_k^\Sigma$, tenemos que $M(\bar{P}) \in \mathrm{R}_{k+1}^\Sigma$ y por lo tanto $M(\bar{P}) \in \mathrm{R}^\Sigma$.

(b) Primero veremos que $D_{M(\bar{P})}$ es un conjunto Σ-p.r.. Notese que

$\chi_{D_{M(\bar{P})}} = \lambda\vec{x}\vec{\alpha}\left[(\exists t \in \omega)_{t \leq f(\vec{x},\vec{\alpha})}\ \bar{P}(t,\vec{x},\vec{\alpha})\right]$

lo cual nos dice que $\chi_{D_{M(\bar{P})}} = \lambda x\vec{x}\vec{\alpha}\left[(\exists t \in \omega)_{t \leq x}\ \bar{P}(t,\vec{x},\vec{\alpha})\right] \circ (f, p_1^{n,m}, ..., p_{n+m}^{n,m})$

Pero el Lema 39 nos dice que $\lambda x\vec{x}\vec{\alpha}\left[(\exists t \in \omega)_{t \leq x}\ \bar{P}(t,\vec{x},\vec{\alpha})\right]$ es Σ-p.r. por lo cual tenemos que $\chi_{D_{M(\bar{P})}}$ lo es. Sea

$P_1 = \lambda t\vec{x}\vec{\alpha}\left[\bar{P}(t,\vec{x},\vec{\alpha}) \wedge (\forall j \in \omega)_{j \leq t}\ j = t \vee \neg\bar{P}(j,\vec{x},\vec{\alpha})\right]$

Note que P_1 es Σ-total. Dejamos al lector usando lemas anteriores probar que P_1 es Σ-p.r.. Ademas notese que para cada $(\vec{x},\vec{\alpha}) \in \omega^n \times \Sigma^{*m}$ tenemos que $P_1(t,\vec{x},\vec{\alpha}) = 1$ si y solo si $t = M(\bar{P})(\vec{x},\vec{\alpha})$

Esto nos dice que $M(\bar{P}) = \left(\lambda\vec{x}\vec{\alpha}\left[\prod_{t=0}^{f(\vec{x},\vec{\alpha})} t^{P_1(t,\vec{x},\vec{\alpha})}\right]\right)|_{D_{M(\bar{P})}}$

por lo cual para probar que $M(\bar{P})$ es Σ-p.r. solo nos resta probar que $F = \lambda\vec{x}\vec{\alpha}\left[\prod_{t=0}^{f(\vec{x},\vec{\alpha})} t^{P_1(t,\vec{x},\vec{\alpha})}\right]$

lo es. Pero $F = \lambda xy\vec{x}\vec{\alpha}\left[\prod_{t=x}^{y} t^{P_1(t,\vec{x},\vec{\alpha})}\right] \circ (C_0^{n,m}, f, p_1^{n,m}, ..., p_{n+m}^{n,m})$

y por lo tanto el Lema 38 nos dice que F es Σ-p.r.. De esta manera hemos probado que $M(\bar{P})$ es Σ-p.r. y por lo tanto $M(P)$ lo es. $\Box$ El lema de minimizacion recien probado es muy util como veremos en los siguientes dos lemas.

Lema 44 Las siguientes funciones son $\varnothing$-p.r.: (a) $\begin{array}{rcl} Q : \omega \times \mathbf{N} & \rightarrow & \omega \\ (x,y) & \rightarrow & \text{cociente de la division de } x \text{ por } y \end{array}$

(b) $\begin{array}{rcl} R : \omega \times \mathbf{N} & \rightarrow & \omega \\ (x,y) & \rightarrow & \text{resto de la division de } x \text{ por } y \end{array}$ (c) $\begin{array}{rcl} pr : \mathbf{N} & \rightarrow & \omega \\ n & \rightarrow & n\text{-esimo numero primo} \end{array}$

Prueba: (a) Veamos primero veamos que $Q = M(P)$, donde $P = \lambda txy\left[(t+1).y > x\right]$. Notar que

$$\begin{array}{rcl} D_{M(P)} & = & \{(x,y) : (\exists t \in \omega)\ P(t,x,y) = 1\} \\ & = & \{(x,y) : (\exists t \in \omega)\ (t+1).y > x\} \\ & = & \omega \times \mathbf{N} \\ & = & D_Q \end{array}$$

Dejamos al lector la facil verificacion de que para cada $(x,y) \in \omega \times \mathbf{N}$, se tiene que $Q(x,y) = M(P)(x,y) = \min\limits_t (t+1).y > x$

Esto prueba que $Q = M(P)$. Ya que P es $\varnothing$-p.r. y $Q(x,y) \leq p_1^{2,0}(x,y)$, para cada $(x,y) \in \omega \times \mathbf{N}$

(b) del Lema 43 implica que $Q \in \mathrm{PR}^\varnothing$. (b) Notese que

$R = \lambda xy\left[x \dot{-} Q(x,y).y\right]$

y por lo tanto $R \in \mathrm{PR}^\varnothing$. (c) Para ver que pr es $\varnothing$-p.r., veremos que la extension $h : \omega \rightarrow \omega$, dada por $h(0) = 0$ y $h(n) = pr(n)$, $n \geq 1$, es $\varnothing$-p.r.. Primero note que

$$\begin{array}{rcl} h(0) & = & 0 \\ h(x+1) & = & \min_t (t \text{ es primo} \wedge t > h(x)) \end{array}$$

O sea que $h = R\left(C_0^{0,0}, M(P)\right)$, donde $P = \lambda tzx\left[t \text{ es primo} \wedge t > z\right]$

Es decir que solo nos resta ver que $M(P)$ es $\varnothing$-p.r.. Claramente P es $\varnothing$-p.r.. Veamos que para cada $(z,x) \in \omega^2$, tenemos que $M(P)(z,x) = \min_t (t \text{ es primo} \wedge t > z) \leq z! + 1$

Sea p primo tal que p divide a $z!+1$. Es facil ver que entonces $p > z$. Pero esto claramente nos dice que $\min_t (t \text{ es primo} \wedge t > z) \leq p \leq z! + 1$

O sea que (b) del Lema 43 implica que $M(P)$ es $\varnothing$ -p.r. ya que podemos tomar $f = \lambda zx\,[z!+1]$. $\square$

Lema 45 Las funciones $\lambda xi\,[(x)_i]$ y $\lambda x\,[Lt(x)]$ son $\varnothing$-p.r. Prueba: Note que $D_{\lambda xi[(x)_i]} = \mathbf{N}\times\mathbf{N}$. Sea

$P = \lambda txi\left[\neg(pr(i)^{t+1} \text{ divide } x)\right]$

Note que P es $\varnothing$-p.r. y que $D_P = \omega\times\omega\times\mathbf{N}$. Dejamos al lector la prueba de que $\lambda xi\,[(x)_i] = M(P)$. Ya que $(x)_i \leq x$, para todo $x\in\mathbf{N}$, (b) del Lema 43 implica que $\lambda xi\,[(x)_i]$ es $\varnothing$-p.r.. Veamos que $\lambda x\,[Lt(x)]$ es $\varnothing$-p.r.. Sea

$Q = \lambda tx\,[(\forall i\in\mathbf{N})_{i\leq x}\ (i\leq t \vee (x)_i = 0)]$

Notese que $D_Q = \omega\times\mathbf{N}$ y que ademas por el Lema 39 tenemos que Q es $\varnothing$-p.r. (dejamos al lector explicar como se aplica tal lema en este caso). Ademas notese que $\lambda x\,[Lt(x)] = M(Q)$ y que $Lt(x)\leq x,$ para todo $x\in\mathbf{N}$

lo cual por (b) del Lema 43 nos dice que $\lambda x\,[Lt(x)]$ es $\varnothing$-p.r.. $\square$ Para $x_1,...,x_n\in\omega$, escribiremos $\langle x_1,...,x_n\rangle$ en lugar de $\langle x_1,...,x_n,0,...\rangle$.

Lema 46 Sea $n\geq 1$. La funcion $\lambda x_1...x_n\,[\langle x_1,...,x_n\rangle]$ es $\varnothing$-p.r. Prueba: Sea $f_n = \lambda x_1...x_n\,[\langle x_1,...,x_n\rangle]$. Claramente f_1 es $\varnothing$ -p.r.. Ademas note que para cada $n\geq 1$, tenemos

$f_{n+1} = \lambda x_1...x_{n+1}\left[(f_n(x_1,...,x_n)pr(n+1)^{x_{n+1}})\right].$

O sea que podemos aplicar un argumento inductivo. $\square$

Minimizacion de variable alfabetica

Supongamos que $\Sigma\neq\varnothing$. Sea $<$ un orden total estricto sobre Σ. Recordemos que $<$ puede ser naturalmente extendido a un orden total estricto sobre Σ^*. Sea $P: D_P\subseteq\omega^n\times\Sigma^{*m}\times\Sigma^*\to\omega$ un predicado. Cuando $(\vec{x},\vec{\alpha})\in\omega^n\times\Sigma^{*m}$ es tal que existe al menos un $\alpha\in\Sigma^*$ tal que $P(\vec{x},\vec{\alpha},\alpha)=1$, usaremos $\min_\alpha^< P(\vec{x},\vec{\alpha},\alpha)$ para denotar al menor $\alpha\in\Sigma^*$ tal que $P(\vec{x},\vec{\alpha},\alpha)=1$. Definamos una funcion

$M^<(P): D_{M^<(P)}\subseteq\omega^n\times\Sigma^{*m}\to\omega$

de la siguiente manera $\begin{array}{rcl} D_{M^<(P)} & = & \{(\vec{x},\vec{\alpha})\in\omega^n\times\Sigma^{*m} : (\exists\alpha\in\Sigma^*)\ P(\vec{x},\vec{\alpha},\alpha)\} \\ M^<(P)(\vec{x},\vec{\alpha}) & = & \min_\alpha^< P(\vec{x},\vec{\alpha},\alpha), \text{ para cada } (\vec{x},\vec{\alpha})\in D_{M^<(P)} \end{array}$

Es decir $M^<(P)$ es exactamente la funcion $\lambda\vec{x}\vec{\alpha}\left[\min_\alpha^< P(\vec{x},\vec{\alpha},\alpha)\right]$. Diremos que $M^<(P)$ es obtenida por minimizacion de variable alfabetica a partir de P.

Lema 47 Supongamos que $\Sigma\neq\varnothing$. Sea $<$ un orden total estricto sobre Σ, sean $n,m\geq 0$ y sea $P: D_P\subseteq\omega^n\times\Sigma^{*m}\times\Sigma^*\to\omega$ un predicado Σ-p.r.. Entonces (a) $M^<(P)$ es Σ-recursiva. (b) Si existe una funcion Σ-p.r. $f:\omega^n\times\Sigma^{*m}\to\omega$ tal que $|M^<(P)(\vec{x},\vec{\alpha})| = \left|\min_\alpha^< P(\vec{x},\vec{\alpha},\alpha)\right| \leq f(\vec{x},\vec{\alpha})$, para cada $(\vec{x},\vec{\alpha})\in D_{M^<(P)}$, entonces $M^<(P)$ es Σ-p.r.. Prueba: Sea $Q = P\circ\left(p_2^{1+n,m},...,p_{1+n+m}^{1+n,m},*^<\circ p_1^{1+n,m}\right)$. Note que

$M^<(P) = *^<\circ M(Q)$

lo cual por (a) del Lema 43 implica que $M^<(P)$ es Σ-recursiva. Sea k el cardinal de Σ. Ya que

$|*^<(M(Q)(\vec{x},\vec{\alpha}))| = |M^<(P)(\vec{x},\vec{\alpha})| \leq f(\vec{x},\vec{\alpha}),$

para todo $(\vec{x},\vec{\alpha})\in D_{M^<(P)} = D_{M(Q)}$, tenemos que $M(Q)(\vec{x},\vec{\alpha}))\leq \sum_{\iota=1}^{i=f(\vec{x},\vec{\alpha})} k^i$, para cada $(\vec{x},\vec{\alpha})\in D_{M(Q)}$.

O sea que por (a) del Lema 43, $M(Q)$ es Σ-p.r. y por lo tanto $M^<(P)$ lo es. $\square$

0.3.3. Recursion primitiva sobre valores anteriores

Dada una funcion $h:\omega\times U\rightarrow\omega$ con $U\subseteq\omega^{n}\times\Sigma^{*m}$, definamos $h^{\downarrow}:\omega\times U\rightarrow\omega$ de la siguiente manera

$$\begin{array}{rcl} h^{\downarrow}(x,\vec{x},\vec{\alpha}) & = & \langle h(0,\vec{x},\vec{\alpha}),h(1,\vec{x},\vec{\alpha}),...,h(x,\vec{x},\vec{\alpha})\rangle \\ & = & \Pi_{i=0}^{x}pr(i+1)^{h(i,\vec{x},\vec{\alpha})} \end{array}$$

Lema 48 Supongamos $\begin{array}{rcl} f & : & U\subseteq\omega^{n}\times\Sigma^{*m}\rightarrow\omega \\ g & : & \omega\times\omega\times U\rightarrow\omega \\ h & : & \omega\times U\rightarrow\omega \end{array}$

son funciones tales que $\begin{array}{rcl} h(0,\vec{x},\vec{\alpha}) & = & f(\vec{x},\vec{\alpha})\text{, para cada }(\vec{x},\vec{\alpha})\in U \\ h(x+1,\vec{x},\vec{\alpha}) & = & g(h^{\downarrow}(x,\vec{x},\vec{\alpha}),x,\vec{x},\vec{\alpha})\text{, para cada }x\in\omega\text{ y }(\vec{x},\vec{\alpha})\in U. \end{array}$

Entonces h es Σ-p.r. si f y g lo son. Prueba: Supongamos f,g son Σ-p.r.. Primero veremos que $h^{\downarrow}$ es Σ-p.r.. Notese que

$$\begin{array}{rcl} h^{\downarrow}(0,\vec{x},\vec{\alpha}) & = & \langle h(0,\vec{x},\vec{\alpha})\rangle \\ & = & \langle f(\vec{x},\vec{\alpha})\rangle \\ & = & 2^{f(\vec{x},\vec{\alpha})} \\ h^{\downarrow}(x+1,\vec{x},\vec{\alpha}) & = & h^{\downarrow}(x,\vec{x},\vec{\alpha})pr(x+2)^{h(x+1,\vec{x},\vec{\alpha})} \\ & = & h^{\downarrow}(x,\vec{x},\vec{\alpha})pr(x+2)^{g(h^{\downarrow}(x,\vec{x},\vec{\alpha}),x,\vec{x},\vec{\alpha})} \end{array}$$

lo cual nos dice que $h^{\downarrow}=R(f_1,g_1)$ donde $\begin{array}{rcl} f_1 & = & \lambda\vec{x}\vec{\alpha}\left[2^{f(\vec{x},\vec{\alpha})}\right] \\ g_1 & = & \lambda Ax\vec{x}\vec{\alpha}\left[Apr(x+2)^{g(A,x,\vec{x},\vec{\alpha})}\right] \end{array}$

O sea que $h^{\downarrow}$ es Σ-p.r. ya que f_1 y g_1 lo son. Finalmente notese que $h=\lambda ix[(x)_i]\circ(Suc\circ p_1^{1+n,m},h^{\downarrow})$

lo cual nos dice que h es Σ-p.r.. $\Box$

0.3.4. Independencia del alfabeto

Probaremos que los conceptos de Σ-recursividad y Σ -recursividad primitiva son en realidad independientes del alfabeto Σ , es decir que si f es una funcion la cual es Σ-mixta y Γ -mixta, entonces f es Σ-recursiva (resp. Σ-p.r.) sii f es Γ-recursiva (resp. Γ-p.r.). Necesitaremos tres lemas.

Lema 49 Supongamos $\varnothing\neq\Sigma\subseteq\Gamma$. (a) Si $<$ es un orden total estricto sobre Σ, entonces las funciones $*^{<}:\omega\rightarrow\Sigma^{*}$ y $\#^{<}:\Sigma^{*}\rightarrow\omega$ son Γ-p.r.. (b) Si $\prec$ es un orden total estricto sobre Γ, entonces las funciones $\#^{\prec}|_{\Sigma^{*}}:\Sigma^{*}\rightarrow\omega$ y $*^{\prec}|_{\#^{\prec}(\Sigma^{*})}:\#^{\prec}(\Sigma^{*})\rightarrow\Sigma^{*}$ son Σ-p.r.. Prueba: (a) Supongamos $\Sigma=\{a_1,...,a_k\}$ y $<$ es dado por $a_1<...<a_k$. Sea $s_e^{<}:\Gamma^{*}\rightarrow\Gamma^{*}$ dada por

$$\begin{array}{rcl} s_e^{<}(\varepsilon) & = & a_1 \\ s_e^{<}(\alpha a_i) & = & \alpha a_{i+1}\text{, si }i<k \\ s_e^{<}(\alpha a_k) & = & s_e^{<}(\alpha)a_1 \\ s_e^{<}(\alpha a) & = & \varepsilon\text{, si }a\in\Gamma-\Sigma. \end{array}$$

Note que $s_e^{<}$ es Γ-p.r. y que $s_e^{<}|_{\Sigma^{*}}=s^{<}$. Ya que Σ^{*} es un conjunto Γ-p.r. tenemos que $s^{<}$ es Γ-p.r.. O sea que la recursion $\begin{array}{rcl} *^{<}(0) & = & \varepsilon \\ *^{<}(x+1) & = & s^{<}(*^{<}(x)) \end{array}$

implica que $*^{<}$ es Γ-p.r.. Para ver que $\#^{<}:\Sigma^{*}\rightarrow\omega$ es Γ-p.r., sea $\#_e^{<}:\Gamma^{*}\rightarrow\omega$ dada por

$$\begin{array}{rcl} \#_e^{<}(\varepsilon) & = & 0 \\ \#_e^{<}(\alpha a_i) & = & \#_e^{<}(\alpha).k+i \\ \#_e^{<}(\alpha a) & = & 0\text{, si }a\in\Gamma-\Sigma. \end{array}$$

Ya que $\#_e^{<}$ es Γ-p.r., eso es $\#^{<}=\#_e^{<}|_{\Sigma^{*}}$. (b) Sea n el cardinal de Γ. Ya que

$\#^{\prec}|_{\Sigma^{*}}(\varepsilon)=0$

$\#^{\prec}|_{\Sigma^{*}}(\alpha a)=\#^{\prec}|_{\Sigma^{*}}(\alpha).n+\#^{\prec}(a)$, para cada $a\in\Sigma$

la funcion $\#^{\prec}|_{\Sigma^{*}}$ es Σ-p.r.. O sea que el predicado $P=\lambda x\alpha\left[\#^{\prec}|_{\Sigma^{*}}(\alpha)=x\right]$ es Σ-p.r.. Sea $<$ un orden total estricto sobre Σ. Note que $*^{\prec}|_{\#^{\prec}(\Sigma^{*})}=M^{<}(P)$, lo cual ya que $\left|*^{\prec}|_{\#^{\prec}(\Sigma^{*})}(x)\right|\leq x$

nos dice que $*^{\prec}|_{\#^{\prec}(\Sigma^*)}$ es Σ-p.r. (Lema 47). $\square$ Supongamos $\Sigma \neq \varnothing$ y sea $<$ un orden total estricto sobre Σ. Para $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$, definamos

$f^{\#^<} = f \circ \left(p_1^{n+m,0}, ..., p_n^{n+m,0}, *^< \circ p_{n+1}^{n+m,0}, ..., *^< \circ p_{n+m}^{n+m,0}\right)$.

Similarmente, para $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$, definamos $f^{\#^<} = \#^< \circ f \circ \left(p_1^{n+m,0}, ..., p_n^{n+m,0}, *^< \circ p_{n+1}^{n+m,0}, ...\right.$

Lema 50 Supongamos $\Gamma \neq \varnothing$ y sea $<$ un orden total estricto sobre Γ. Dada h una funcion Γ-mixta, son equivalentes (1) h es Γ-recursiva (resp. Γ-p.r.) (2) $h^{\#^<}$ es $\varnothing$-recursiva (resp. $\varnothing$-p.r.)
Prueba: (2)$\Rightarrow$(1). Supongamos $h : D_h \subseteq \omega^n \times \Gamma^{*m} \to \Gamma^*$. Ya que $h^{\#^<}$ es Γ -recursiva (resp. Γ-p.r.) y

$h = *^< \circ h^{\#^<} \circ (p_1^{n,m}, ..., p_n^{n,m}, \#^< \circ p_{n+1}^{n,m}, ..., \#^< \circ p_{n+m}^{n,m})$,

tenemos que h es Γ-recursiva (resp. Γ-p.r.). (1)$\Rightarrow$(2). Probaremos por induccion en k que

(*) Si $h \in \mathrm{R}_k^\Gamma$ (resp. $h \in \mathrm{PR}_k^\Gamma$), entonces $h^{\#^<}$ es $\varnothing$-recursiva (resp. $\varnothing$-p.r.). El caso $k = 0$ es facil y dejado al lector. Supongamos (*) vale para un k fijo. Veremos que vale para $k+1$. Sea $h \in \mathrm{R}_{k+1}^\Gamma$ (resp. $h \in \mathrm{PR}_{k+1}^\Gamma$). Hay varios casos

Caso 1. Supongamos $h = f \circ (f_1, ..., f_n)$, con $f, f_1, ..., f_n \in \mathrm{R}_k^\Gamma$ (resp. $f, f_1, ..., f_n \in \mathrm{PR}_k^\Gamma$). Por hipotesis inductiva tenemos que $f^{\#^<}, f_1^{\#^<}, ..., f_n^{\#^<}$ son $\varnothing$-recursivas (resp. $\varnothing$ -p.r.). Ya que $h^{\#^<} = f^{\#^<} \circ \left(f_1^{\#^<}, ..., f_n^{\#^<}\right)$, tenemos que $h^{\#^<}$ es $\varnothing$-recursiva (resp. $\varnothing$-p.r.).

Caso 2. Supongamos $h = M(P)$, con $P : \omega \times \omega^n \times \Gamma^{*m} \to \omega$, un predicado en R_k^Γ. Ya que $h^{\#^<} = M(P^{\#^<})$, tenemos que $h^{\#^<}$ es $\varnothing$ -recursiva.

Caso 3. Supongamos $h = R(f, \mathcal{G})$, con

$$\begin{array}{lll} f & : & \omega^n \times \Gamma^{*m} \to \Gamma^* \\ \mathcal{G}_a & : & \omega^n \times \Gamma^{*m} \times \Gamma^* \times \Gamma^* \to \Gamma^*,\ a \in \Gamma \end{array}$$

funciones en R_k^Γ (resp. PR_k^Γ). Sea $\Gamma = \{a_1, ..., a_r\}$, con $a_1 < a_2 < ... < a_r$. Por hipotesis inductiva tenemos que $f^{\#^<}$ y cada $\mathcal{G}_a^{\#^<}$ son $\varnothing$-recursivas (resp. $\varnothing$-p.r.). Sea

$$\begin{array}{rcl} i_0 : \omega & \to & \omega \\ x & \to & \left\{\begin{array}{ll} r & \text{si } r \text{ divide } x \\ R(x,r) & \text{caso contrar} \end{array}\right. \end{array}$$

y sea $B = \lambda x \left[Q(x \dot{-} i_0(x), r)\right]$

(R y Q son definidas en el Lema 44). Note que i_0 y B son $\varnothing$-p.r. y que $*^<(x) = *^<(B(x))a_{i_0(x)}$, para $x \geq 1$

(ver Lema 6). Tambien tenemos

$$\begin{array}{rcl} h^{\#^<}(\vec{x}, \vec{y}, t+1) & = & \#^<(h(\vec{x}, *^<(\vec{y}), *^<(t+1))) \\ & = & \#^<(h(\vec{x}, *^<(\vec{y}), *^<(B(t+1))a_{i_0(t+1)})) \\ & = & \#^<\left(\mathcal{G}_{a_{i_0(t+1)}}(\vec{x}, *^<(\vec{y}), *^<(B(t+1)), h(\vec{x}, *^<(\vec{y}), *^< \right. \\ & = & \#^<\left(\mathcal{G}_{a_{i_0(t+1)}}(\vec{x}, *^<(\vec{y}), *^<(B(t+1)), *^<(h^{\#^<}(\vec{x}, \vec{y}, \right. \\ & = & \mathcal{G}^{\#^<}_{a_{i_0(t+1)}}(\vec{x}, \vec{y}, B(t+1), h^{\#^<}(\vec{x}, \vec{y}, B(t+1))) \end{array}$$

y ya que $B(t+1) < t+1$, tenemos que (**) $h^{\#^<}(\vec{x}, \vec{y}, t+1) = \mathcal{G}^{\#^<}_{a_{i_0(t+1)}}(\vec{x}, \vec{y}, B(t+1), \left\langle h^{\#^<}(\vec{x}, \vec{y}, 0), ..., h^{\#^<}(\vec{x}\right.$

A continuacion definamos

$H = \lambda t\vec{x}\vec{y}\left[\left\langle h^{\#^<}(\vec{x}, \vec{y}, 0), ..., h^{\#^<}(\vec{x}, \vec{y}, t)\right\rangle\right]$

Por (**) tenemos que

$$\begin{array}{rcl} H(0, \vec{x}, \vec{y}) & = & \left\langle h^{\#^<}(\vec{x}, \vec{y}, 0)\right\rangle = \left\langle f^{\#^<}(\vec{x}, \vec{y})\right\rangle = 2^{f^{\#^<}(\vec{x},\vec{y})} \\ H(t+1, \vec{x}, \vec{y}) & = & \left((H(t, \vec{x}, \vec{y}) + 1).pr(t+2)^{\mathcal{G}^{\#^<}_{a_{i_0(t+1)}}(\vec{x},\vec{y},B(t+1),(H(t,\vec{x},\vec{y}))_{B(t+1)})}\right) \end{array}$$

O sea que si definimos $g : \omega \times \omega \times \omega^n \times \omega^m \to \omega$ por $g(z, t, \vec{x}, \vec{y}) = \left\{\begin{array}{l} \left((z+1).pr(t+2)^{\mathcal{G}^{\#^<}_{a_1}(\vec{x},\vec{y},B(t+1),(z)_{B(t+1}} \right. \\ \vdots \\ \left((z+1).pr(t+2)^{\mathcal{G}^{\#^<}_{a_r}(\vec{x},\vec{y},B(t+1),(z)_{B(t+1}} \right. \end{array}\right.$

tenemos que $H = R(\lambda x [2^x] \circ f^{\#^<}, g)$. Note que g es $\varnothing$-recursiva (resp. $\varnothing$-p.r.), ya que $g = f_1(z, t, \vec{x}, \vec{y})P_1(z, t, \vec{x}, \vec{y}) + ... + f_r(z, t, \vec{x}, \vec{y})P_r(z, t, \vec{x}, \vec{y})$, con $\begin{array}{rcl} f_i & = & \lambda zt\vec{x}\vec{y}\left[\left((z+1).pr(t+2)^{\mathcal{G}_{a_i}^{\#^<}(\vec{x},\vec{y},B(t+1),(z)_{B(t+1)})}\right)\right] \\ P_i & = & \lambda zt\vec{x}\vec{y}\left[i_0(t+1)=i\right] \end{array}$

y estas funciones son totales y $\varnothing$-recursivas (resp. $\varnothing$ -p.r.). O sea que H es $\varnothing$-recursiva (resp. $\varnothing$-p.r.) y por lo tanto lo es $h^{\#^<} = \lambda\vec{x}\vec{y}t\left[(H(t,\vec{x},\vec{y}))_{t+1}\right]$

Los otros casos en los cuales h es obtenida por recursion primitiva son similares. $\square$ Ahora podemos probar el anunciado resultado de independencia.

Teorema 51 Sean Σ y Γ alfabetos cualesquiera. (a) Supongamos una funcion f es Σ-mixta y Γ-mixta, entonces f es Σ-recursiva (resp. Σ-p.r.) sii f es Γ-recursiva (resp. Γ-p.r.). (b) Supongamos un conjunto S es Σ-mixto y Γ-mixto, entonces S es Σ-p.r. sii S es Γ-p.r.. Prueba: (a) Ya que f es $(\Sigma\cap\Gamma)$-mixta, podemos suponer sin perdida de generalidad que $\Sigma\subseteq\Gamma$. Primero haremos el caso en que $\Sigma=\varnothing$ y $\Gamma\neq\varnothing$. Sea $<$ un orden total estricto sobre Γ. Ya que f es $\varnothing$-mixta, tenemos $f=f^{\#^<}$ y por lo tanto podemos aplicar el lema anterior.

Supongamos ahora que $\Sigma\neq\varnothing$. O sea que $f: D_f\subseteq\omega^n\times\Sigma^{*m}\to O$, con $O\in\{\omega,\Sigma^*\}$. Haremos el caso $O=\Sigma^*$. Supongamos f es Σ -recursiva (resp. Σ-p.r.). Sea $\prec$ un orden total estricto sobre Γ. Ya que las funciones $\#^\prec|_{\Sigma^*}$ y $*^\prec|_{\#^\prec(\Sigma^*)}$ son Σ-p.r. (Lema 49) y

$$\begin{array}{rcl} f^{\#^\prec} & = & \#^\prec\circ f\circ\left(p_1^{n+m,0},...,p_n^{n+m,0},*^\prec\circ p_{n+1}^{n+m,0},...,*^\prec\circ p_{n+m}^{n+m,0}\right) \\ & = & \#^\prec|_{\Sigma^*}\circ f\circ\left(p_1^{n+m,0},...,p_n^{n+m,0},*^\prec|_{\#^\prec(\Sigma^*)}\circ p_{n+1}^{n+m,0},...,*^\prec|_{\#^\prec(\Sigma^*)}\circ p_{n+m}^{n+m,0}\right) \end{array}$$

tenemos que $f^{\#^\prec}$ es Σ-recursiva (resp. Σ-p.r.). O sea que por el caso ya probado de (a), $f^{\#^\prec}$ es $\varnothing$ -recursiva (resp. $\varnothing$-p.r.) lo cual por el lema anterior nos dice que f es Γ-recursiva (resp. Γ-p.r.). Supongamos ahora que f es Γ-recursiva (resp. Γ-p.r.). Sea $<$ un orden total estricto sobre Σ. Ya que $\#^<$ y $*^<$ son Γ-p.r. (Lema 49), la funcion

$$f^{\#^<}=\#^<\circ f\circ\left(p_1^{n+m,0},...,p_n^{n+m,0},*^<\circ p_{n+1}^{n+m,0},...,*^<\circ p_{n+m}^{n+m,0}\right)$$

es Γ-recursiva (resp. Γ-p.r.). Por el caso ya probado de (a), $f^{\#^<}$ es $\varnothing$-recursiva (resp. $\varnothing$-p.r.), lo cual por el lema anterior nos dice que f es Σ-recursiva (resp. Σ-p.r.). (b) es dejado al lector (use (a)). $\square$

0.4. El lenguaje $\mathcal{S}^{\Sigma}$

En esta seccion introducimos un lenguaje de programacion teorico el cual depende de un alfabeto Σ previamente fijado. Este lenguaje, llamado $\mathcal{S}^{\Sigma}$, nos servira para dar una version imperativa del concepto de funcion Σ-efectivamente computable.

0.4.1. Sintaxis de $\mathcal{S}^{\Sigma}$

Necesitaremos algunas funciones basicas para poder describir la sintaxis de $\mathcal{S}^{\Sigma}$ en forma precisa. Llamaremos numerales a los siguientes simbolos

0 1 2 3 4 5 6 7 8 9

Usaremos Num para denotar el conjunto de numerales. Notese que $Num \cap \omega = \varnothing$. Sea

$S : Num^* \to Num^*$ definida de la siguiente manera
$$\begin{array}{rcl} S(\varepsilon) & = & 1 \\ S(\alpha 0) & = & \alpha 1 \\ S(\alpha 1) & = & \alpha 2 \\ S(\alpha 2) & = & \alpha 3 \\ S(\alpha 3) & = & \alpha 4 \\ S(\alpha 4) & = & \alpha 5 \\ S(\alpha 5) & = & \alpha 6 \\ S(\alpha 6) & = & \alpha 7 \\ S(\alpha 7) & = & \alpha 8 \\ S(\alpha 8) & = & \alpha 9 \\ S(\alpha 9) & = & S(\alpha)0 \end{array}$$

Definamos $\overline{\quad} : \omega \to Num^*$ de la siguiente manera
$$\begin{array}{rcl} \bar{0} & = & \varepsilon \\ \overline{n+1} & = & S(\bar{n}) \end{array}$$

Notese que para $n \in \mathbf{N}$, la palabra $\bar{n}$ es la notacion usual decimal de n. Para $\alpha \in \Sigma^*$, sea

$$^{\frown}\alpha = \left\{ \begin{array}{ll} [\alpha]_2 \ldots [\alpha]_{|\alpha|} & \text{si } \ |\alpha| \geq 2 \\ \varepsilon & \text{si } \ |\alpha| \leq 1 \end{array} \right.$$

La sintaxis de $\mathcal{S}^{\Sigma}$ sera dada utilizando solo simbolos del alfabeto $\Sigma \cup \Sigma_p$, donde $\Sigma_p = Num \cup \{\leftarrow, +, \dot{-}, ., \neq, ^{\frown}, \varepsilon, \mathrm{N}, \mathrm{K}, \mathrm{P}, \mathrm{L}, \mathrm{I}, \mathrm{F}, \mathrm{G}, \mathrm{O}, \mathrm{T}, \mathrm{B}, \mathrm{E}, \mathrm{S}\}$.

Cabe aclarar que la palabra de longitud 0 no es un elemento de Σ_p sino que la letra griega ε que usualmente denota esta palabra, lo es. Las palabras de la forma $\mathrm{N}\bar{k}$ con $k \in \mathbf{N}$, son llamadas variables numericas de $\mathcal{S}^{\Sigma}$. Las palabras de la forma $\mathrm{P}\bar{k}$ con $k \in \mathbf{N}$, son llamadas variables alfabeticas de $\mathcal{S}^{\Sigma}$. Las palabras de la forma $\mathrm{L}\bar{k}$ con $k \in \mathbf{N}$, son llamadas labels de $\mathcal{S}^{\Sigma}$. Una instruccion basica de $\mathcal{S}^{\Sigma}$ es un elemento de $(\Sigma \cup \Sigma_p)^*$ el cual es de alguna de las siguientes formas

$\mathrm{N}\bar{k} \leftarrow \mathrm{N}\bar{k} + 1$ $\mathrm{N}\bar{k} \leftarrow \mathrm{N}\bar{k} \dot{-} 1$ $\mathrm{N}\bar{k} \leftarrow \mathrm{N}\bar{n}$ $\mathrm{N}\bar{k} \leftarrow 0$ $\mathrm{P}\bar{k} \leftarrow \mathrm{P}\bar{k}.a$ $\mathrm{P}\bar{k} \leftarrow {}^{\frown}\mathrm{P}\bar{k}$ $\mathrm{P}\bar{k} \leftarrow \mathrm{P}\bar{n}$ $\mathrm{P}\bar{k} \leftarrow \varepsilon$ IF $\mathrm{N}\bar{k} \neq 0$ GOTO $\mathrm{L}\bar{n}$ IF $\mathrm{P}\bar{k}$ BEGINS a GOTO $\mathrm{L}\bar{n}$ GOTO $\mathrm{L}\bar{n}$ SKIP donde $a \in \Sigma$ y $k, n \in \mathbf{N}$. Como puede observarse para que las instrucciones basicas sean mas lejibles usaremos espacios entre ciertos simbolos. Por ejemplo, escribiremos

L1 IF N5 $\neq$ 0 GOTO L3

en lugar de L1IFN5$\neq$0GOTOL3

pero debe entenderse que la instruccion basica a la que nos referimos es esta ultima palabra de longitud 14. Una instruccion de $\mathcal{S}^{\Sigma}$ es una palabra de la forma αI, donde $\alpha \in \{\mathrm{L}\bar{n} : n \in \mathbf{N}\} \cup \{\varepsilon\}$ y I es una instruccion basica. Usaremos Ins^{Σ} para denotar el conjunto de todas las instrucciones de $\mathcal{S}^{\Sigma}$. Cuando la instruccion I es de la forma $\mathrm{L}\bar{n}J$ con J una instruccion basica, diremos que $\mathrm{L}\bar{n}$ es el label de I. Damos a continuacion, a modo de ejemplo, la interpretacion intuitiva asociada a ciertas instrucciones basicas de $\mathcal{S}^{\Sigma}$:

INSTRUCCION	:	$\mathrm{N}\bar{k} \leftarrow \mathrm{N}\bar{k}\dot{-}1$
INTERPRETACION	:	Si el contenido de $\mathrm{N}\bar{k}$ es 0 dejarlo sin modificar; en caso contrario disminuya en 1 el contenido de $\mathrm{N}\bar{k}$
INSTRUCCION	:	$\mathrm{N}\bar{k} \leftarrow \mathrm{N}\bar{n}$
INTERPRETACION	:	Copiar en $\mathrm{N}\bar{k}$ el contenido de $\mathrm{N}\bar{n}$ sin modificar el contenido de $\mathrm{N}\bar{n}$
INSTRUCTION	:	$\mathrm{P}\bar{k} \leftarrow^{\frown} \mathrm{P}\bar{k}$
INTERPRETATION	:	Si el contenido de $\mathrm{P}\bar{k}$ es ε dejarlo sin modificar; en caso contrario remueva el 1er simbolo del contenido de $\mathrm{P}\bar{k}$
INSTRUCTION	:	$\mathrm{P}\bar{k} \leftarrow \mathrm{P}\bar{k}.a$
INTERPRETATION	:	Modificar el contenido de $\mathrm{P}\bar{k}$ agregandole el simbolo a a la derecha
INSTRUCTION	:	IF $\mathrm{P}\bar{k}$ BEGINS a GOTO $\mathrm{L}\bar{m}$
INTERPRETATION	:	Si el contenido de $\mathrm{P}\bar{k}$ comiensa con a, ejecute la primer instruccion con label $\mathrm{L}\bar{m}$; en caso contrario ejecute la siguiente instruccion

Un programa de $\mathcal{S}^{\Sigma}$ es una palabra de la forma

$I_1 I_2 ... I_n$

donde $n \geq 1$, $I_1, ..., I_n \in \mathrm{Ins}^{\Sigma}$ y para cada $i = 1, ..., n$, tenemos que - si GOTOL$\bar{m}$ es un tramo final de I_i, entonces existe j tal que I_j tiene label L$\bar{m}$ Usaremos Pro^{Σ} para denotar el conjunto de todos los programas de $\mathcal{S}^{\Sigma}$. Como es usual cuando escribamos un programa lo haremos linea por linea, con la finalidad de que sea mas lejible. Por ejemplo, escribiremos

L2 $\quad$ N12 $\leftarrow$ N12$\dot{-}$1
$\quad$ P1 $\leftarrow$ $^{\frown}$P1
$\quad$ IF N12 $\neq$ 0 GOTO L2

en lugar de L2N12$\leftarrow$N12$\dot{-}$1P1$\leftarrow^{\frown}$P1IFN12$\neq$0GOTOL2

Un importante resultado es el siguiente lema que garantiza que los programas pueden ser parseados en forma unica como concatenacion de instrucciones.

Lema 52 Se tiene que: (a) Si $I_1...I_n = J_1...J_m$, con $I_1, ..., I_n, J_1, ..., J_m \in \mathrm{Ins}^{\Sigma}$, entonces $n = m$ y $I_j = J_j$ para cada $j \geq 1$. (b) Si $\mathcal{P} \in \mathrm{Pro}^{\Sigma}$, entonces existe una unica sucesion de instrucciones $I_1, ..., I_n$ tal que $\mathcal{P} = I_1...I_n$ Prueba: (a) Supongamos I_n es un tramo final propio de J_m. Notar que entonces $n > 1$. Es facil ver que entonces ya sea $J_m = \mathrm{L}\bar{u}I_n$ para algun $u \in \mathbf{N}$, o I_n es de la forma GOTO L$\bar{n}$ y J_m es de la forma wIF P$\bar{k}$ BEGINS a GOTO L$\bar{n}$ donde $w \in \{\mathrm{L}\bar{n} : n \in \mathbf{N}\} \cup \{\varepsilon\}$. El segundo caso no puede darse porque entonces el anteultimo simbolo de I_{n-1} deberia ser S lo cual no sucede para ninguna instruccion. O sea que

$I_1...I_n = J_1...J_{m-1}\mathrm{L}\bar{u}I_n$

lo cual dice que (*) $I_1...I_{n-1} = J_1...J_{m-1}\mathrm{L}\bar{u}$. Es decir que L$\bar{u}$ es tramo final de I_{n-1} y por lo tanto GOTO L$\bar{u}$ es tramo final de I_{n-1}. Por (*), GOTO es tramo final de $J_1...J_{m-1}$, lo cual es impossible. Hemos llegado a una contradiccion lo cual nos dice que I_n no es un tramo final propio de J_m. Por simetria tenemos que $I_n = J_m$, lo cual usando un razonamiento inductivo nos dice que $n = m$ y $I_j = J_j$ para cada $j \geq 1$.

(b) Es consecuencia directa de (a). $\Box$

(b) del lema anterior nos dice que dado un programa $\mathcal{P}$, tenemos univocamente determinados $n(\mathcal{P}) \in \mathbf{N}$ y $I_1^{\mathcal{P}}, ..., I_{n(\mathcal{P})}^{\mathcal{P}} \in \mathrm{Ins}^{\Sigma}$ tales que $\mathcal{P} = I_1^{\mathcal{P}}...I_{n(\mathcal{P})}^{\mathcal{P}}$. Definamos tambien

$I_i^{\mathcal{P}} = \varepsilon$

cuando $i = 0$ o $i > n(\mathcal{P})$. O sea que, la funcion $(\Sigma \cup \Sigma_p)$-mixta $\lambda i\mathcal{P}\left[I_i^{\mathcal{P}}\right]$ tiene dominio igual a $\omega \times \mathrm{Pro}^{\Sigma}$. Tambien sera necesaria la funcion $Bas : \mathrm{Ins}^{\Sigma} \rightarrow (\Sigma \cup \Sigma_p)^*$, dada por

$$Bas(I) = \left\{ \begin{array}{ll} J & \text{si } I \text{ es de la forma } \mathrm{L}\bar{k}J \text{ con } J \in \mathrm{Ins}^{\Sigma} \\ I & \text{caso contrario} \end{array} \right.$$

Semantica de $\mathcal{S}^{\Sigma}$

Definamos

$$\begin{array}{rcl} \omega^{[\mathbf{N}]} & = & \left\{(s_1, s_2, ...) \in \omega^{\mathbf{N}} : \text{ hay } n \in \mathbf{N} \text{ tal que } s_i = 0, \text{para } i \geq n\right\} \\ \Sigma^{*[\mathbf{N}]} & = & \left\{(\sigma_1, \sigma_2, ...) \in \Sigma^{*\mathbf{N}} : \text{ hay } n \in \mathbf{N} \text{ tal que } \sigma_i = \varepsilon, \text{para } i \geq n\right\}. \end{array}$$

Asumiremos siempre que en una computacion via un programa de $\mathcal{S}^{\Sigma}$, todas exepto una cantidad finita de las variables numericas tienen el valor 0 y todas exepto una cantiad finita de las variables alfabeticas tienen el valor ε. Esto no quita generalidad a nuestra modelizacion del funcionamiento de los programas ya que todo programa envuelve una cantidad finita de variables. O sea que, en general, independientemente de que programa estemos considerando, un estado sera un par $(\vec{s}, \vec{\sigma}) = ((s_1, s_2, ...), (\sigma_1, \sigma_2, ...)) \in \omega^{[\mathbf{N}]} \times \Sigma^{*[\mathbf{N}]}$.

Si $i \geq 1$, entonces diremos que s_i es el contenido de la variable N$\bar{\imath}$ en el estado $(\vec{s}, \vec{\sigma})$ y σ_i es el contenido de la variable P$\bar{\imath}$ en el estado $(\vec{s}, \vec{\sigma})$. Una descripcion instantanea es una terna $(i, \vec{s}, \vec{\sigma})$ tal que $(\vec{s}, \vec{\sigma})$ es un estado e $i \in \omega$. Dado un programa $\mathcal{P}$ y una descripcion instantanea $(i, \vec{s}, \vec{\sigma})$ definamos la descripcion instantanea sucesora de $(i, \vec{s}, \vec{\sigma})$ como la terna $(j, \vec{u}, \vec{\eta})$ descripta a continuacion en los siguientes casos

Caso $i \notin \{1, ..., n(\mathcal{P})\}$. Entonces $(j, \vec{u}, \vec{\eta}) = (i, \vec{s}, \vec{\sigma})$ (ya que en $(i, \vec{s}, \vec{\sigma})$ el numero i indica que la instruccion $I_i^{\mathcal{P}}$ debe ser ejecutada, es natural definir $(j, \vec{u}, \vec{\eta}) = (i, \vec{s}, \vec{\sigma})$ ya que, en este caso, no se puede ejecutar la instruccion $I_i^{\mathcal{P}} = \varepsilon$).

Caso $Bas(I_i^{\mathcal{P}}) = \text{N}\bar{k} \leftarrow \text{N}\bar{k} \dot{-} 1$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & (s_1, ..., s_{k-1}, s_k \dot{-} 1, s_{k+1}, ...) \\ \vec{\eta} & = & \vec{\sigma} \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) = \text{N}\bar{k} \leftarrow \text{N}\bar{k} + 1$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & (s_1, ..., s_{k-1}, s_k + 1, s_{k+1}, ...) \\ \vec{\eta} & = & \vec{\sigma} \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) = \text{N}\bar{k} \leftarrow \text{N}\bar{n}$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & (s_1, ..., s_{k-1}, s_n, s_{k+1}, ...) \\ \vec{\eta} & = & \vec{\sigma} \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) = \text{N}\bar{k} \leftarrow 0$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & (s_1, ..., s_{k-1}, 0, s_{k+1}, ...) \\ \vec{\eta} & = & \vec{\sigma} \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) = \text{IF N}\bar{k} \neq 0 \text{ GOTO L}\bar{m}$. Entonces

$$\begin{array}{rcl} \vec{u} & = & \vec{s} \\ \vec{\eta} & = & \vec{\sigma} \end{array}$$

y tenemos dos subcasos. Subcaso a. El valor de N$\bar{k}$ en $(\vec{s}, \vec{\sigma})$ es 0. Entonces $j = i + 1$.

Subcaso b. El valor de N$\bar{k}$ en $(\vec{s}, \vec{\sigma})$ es no nulo. Entonces j es el menor numero l tal que la l-esima instruccion de $\mathcal{P}$ tiene label L$\bar{m}$.

Caso $Bas(I_i^{\mathcal{P}}) = \text{P}\bar{k} \leftarrow {}^{\frown}\text{P}\bar{k}$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & \vec{s} \\ \vec{\eta} & = & (\sigma_1, ..., \sigma_{k-1}, {}^{\frown}\sigma_k, \sigma_{k+1}, ...) \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) = \text{P}\bar{k} \leftarrow \text{P}\bar{k}.a$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & \vec{s} \\ \vec{\eta} & = & (\sigma_1, ..., \sigma_{k-1}, \sigma_k a, \sigma_{k+1}, ...) \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) = \text{P}\bar{k} \leftarrow \text{P}\bar{n}$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & \vec{s} \\ \vec{\eta} & = & (\sigma_1,...,\sigma_{k-1},\sigma_n,\sigma_{k+1},...) \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) = \mathrm{P}\bar{k} \leftarrow \varepsilon$. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & \vec{s} \\ \vec{\eta} & = & (\sigma_1,...,\sigma_{k-1},\varepsilon,\sigma_{k+1},...) \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) =$ IF P$\bar{k}$ BEGINS a GOTO L$\bar{m}$. Entonces

$$\begin{array}{rcl} \vec{u} & = & \vec{s} \\ \vec{\eta} & = & \vec{\sigma} \end{array}$$

y tenemos dos subcasos. Subcaso a. El valor de P$\bar{k}$ en $(\vec{s},\vec{\sigma})$ comiensa con a. Entonces j es el menor numero l tal que la l-esima instruccion de $\mathcal{P}$ tiene label L$\bar{m}$.

Subcaso b. El valor de P$\bar{k}$ en $(\vec{s},\vec{\sigma})$ no comiensa con a. Entonces $j = i+1$

Caso $Bas(I_i^{\mathcal{P}}) =$ GOTO L$\bar{m}$. Entonces

$$\begin{array}{rcl} j & = & \text{menor numero } l \text{ tal que la } l\text{-esima instruccion de } \mathcal{P} \text{ tiene label L}\bar{m}. \\ \vec{u} & = & \vec{u} \\ \vec{\eta} & = & \vec{\eta} \end{array}$$

Caso $Bas(I_i^{\mathcal{P}}) =$ SKIP. Entonces

$$\begin{array}{rcl} j & = & i+1 \\ \vec{u} & = & \vec{u} \\ \vec{\eta} & = & \vec{\eta} \end{array}$$

Dado un programa $\mathcal{P}$ y una descripcion instantanea $(i,\vec{s},\vec{\sigma})$ usaremos $DIS_{\mathcal{P}}(i,\vec{s},\vec{\sigma})$ para denotar la descripcion instantanea sucesora de $(i,\vec{s},\vec{\sigma})$ en $\mathcal{P}$. Dado un programa $\mathcal{P}$ y un estado $(\vec{s},\vec{\sigma})$ tenemos asociada una sucesion infinita de descripciones instantaneas

$(i_1,\vec{s}_1,\vec{\sigma}_1),(i_2,\vec{s}_2,\vec{\sigma}_2),(i_3,\vec{s}_3,\vec{\sigma}_3),...$

definida de la siguiente manera $\begin{array}{rcl} (i_1,\vec{s}_1,\vec{\sigma}_1) & = & (1,\vec{s},\vec{\sigma}) \\ (i_{j+1},\vec{s}_{j+1},\vec{\sigma}_{j+1}) & = & DIS_{\mathcal{P}}(i_j,\vec{s}_j,\vec{\sigma}_j) \end{array}$

Diremos que $(\vec{s}_{j+1},\vec{\sigma}_{j+1})$ es el estado obtenido luego de j pasos, partiendo del estado $(\vec{s},\vec{\sigma})$. Tambien diremos que $(i_{j+1},\vec{s}_{j+1},\vec{\sigma}_{j+1})$ es la descripcion instantanea obtenida luego de j pasos, partiendo de la descripcion instantanea $(1,\vec{s},\vec{\sigma})$. La sucesion anterior tiene dos posibles formas Caso 1. Hay un $m \geq 1$ tal que $i_1,...,i_m \leq n(\mathcal{P})$ y $i_{m+k} = n(\mathcal{P})+1$, para cada $k \geq 1$. Caso 2. Para cada $j \geq 1$, tenemos que $i_j \leq n(\mathcal{P})$. Cuando se da el Caso 1, diremos para cada $j \geq m$ que $\mathcal{P}$ se detiene (luego de j pasos), partiendo desde el estado $(\vec{s},\vec{\sigma})$. Si se da el Caso 2 diremos que $\mathcal{P}$ no se detiene partiendo del estado $(\vec{s},\vec{\sigma})$.

Macros

Usaremos como variables numericas de macros a las palabras de la forma V$\bar{m}$, con $m \geq 1$. Usaremos como variables alfabeticas de macros a las palabras de la forma W$\bar{m}$, con $m \geq 1$. Usaremos como variables para labels de macros a las palabras de la forma A$\bar{m}$, con $m \geq 1$. Si $\Sigma = \{@,!,\&\}$ entonces el macro [IF W1 $\neq \varepsilon$ GOTO A1] puede ser hecho de la siguiente manera

IF W1 BEGINS @ GOTO A1
IF W1 BEGINS ! GOTO A1
IF W1 BEGINS & GOTO A1

Para hacer el macro [V1 $\leftarrow$ V2 + V3] podriamos tomar

	V1 $\leftarrow$ V2
	V4 $\leftarrow$ V3
A1	IF V4 $\neq 0$ GOTO A2
	GOTO A3
A2	V4 $\leftarrow$ V4 $-$ 1
	V1 $\leftarrow$ V1 + 1
	GOTO A1
A3	SKIP

Funciones Σ-computables

Dado $\mathcal{P} \in \mathrm{Pro}^{\Sigma}$, definamos para cada par $n, m \geq 0$, la funcion $\Psi_{\mathcal{P}}^{n,m,\omega}$ de la siguiente manera:

$D_{\Psi_{\mathcal{P}}^{n,m,\omega}} = \{(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} : \mathcal{P}$ termina, partiendo del estado $((x_1, ..., x_n, 0, ...), (\alpha_1, ..., \alpha_m, \varepsilon, ...))\}$

$\Psi_{\mathcal{P}}^{n,m,\omega}(\vec{x}, \vec{\alpha}) =$ valor de N1 en el estado obtenido cuando $\mathcal{P}$ termina, partiendo de $((x_1, ..., x_n, 0, ...), (\alpha_1, ..., \alpha_m, \varepsilon, ...))$

Analogamente definamos la funcion $\Psi_{\mathcal{P}}^{n,m,\Sigma^*}$ de la siguiente manera: $D_{\Psi_{\mathcal{P}}^{n,m,\Sigma^*}} = \{(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} : \mathcal{P}$
estado $((x_1$

$\Psi_{\mathcal{P}}^{n,m,\Sigma^*}(\vec{x}, \vec{\alpha}) =$ valor de P1 en el estado obtenido cuando $\mathcal{P}$ termina, partiendo de $(x_1, ..., x_n, 0, ...), (\alpha_1, ..., \alpha_m, \varepsilon, ...))$

Una funcion Σ-mixta $f : S \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ sera llamada Σ-computable si hay un programa $\mathcal{P}$ tal que $f = \Psi_{\mathcal{P}}^{n,m,O}$. En tal caso diremos que la funcion f es computada por $\mathcal{P}$. Ejemplos: (a) El programa

L2	IF N1 $\neq 0$ GOTO L1
	GOTO L2
L1	N1 $\leftarrow$ N1$\dot{-}$1

computa la funcion $Pred$. Note que este programa tambien computa las funciones $Pred \circ p_1^{n,m}$, para $n \geq 1$ y $m \geq 0$. (b) Sea $\Sigma = \{\clubsuit, \triangle\}$. El programa

L3	IF P2 BEGINS $\clubsuit$ GOTO L1
	IF P2 BEGINS $\triangle$ GOTO L2
	GOTO L4
L1	P2 $\leftarrow$ $^\frown$P2
	P1 $\leftarrow$ P1$\clubsuit$
	GOTO L3
L2	P2 $\leftarrow$ $^\frown$P2
	P1 $\leftarrow$ P1$\triangle$
	GOTO L3
L4	SKIP

computa la funcion $\lambda\alpha\beta\,[\alpha\beta]$.

Teorema 53 Si f es Σ-computable, entonces f es Σ-efectivamente computable. Prueba: Supongamos por ejemplo que $f : S \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$ es computada por $\mathcal{P} \in \mathrm{Pro}^{\Sigma}$. Es claro que el procedimiento que consiste en realizar las sucesivas instrucciones de $\mathcal{P}$ (partiendo de $((x_1, ..., x_n, 0, 0, ...), (\alpha_1, ..., \alpha_m, \varepsilon, \varepsilon, ...))$) y eventualmente terminar en caso de que nos toque realizar la instruccion $n(\mathcal{P}) + 1$, y dar como salida el contenido de la variable N1 , es un procedimiento efectivo que computa a f. $\square$

Macros asociados a funciones Σ-computables

Dada una funcion $f : S \subseteq \omega^n \times \Sigma^{*m} \to \omega$, con
$[\mathrm{V}\overline{n+1} \leftarrow f(\mathrm{V}1, ..., \mathrm{V}\bar{n}, \mathrm{W}1, ..., \mathrm{W}\bar{m})]$
denotaremos un macro el cual cumpla lo siguiente. Si reemplazamos sus variables y labels auxiliares por variables y labels concretos (distintos de a dos), y reemplazamos las variables $\mathrm{V}1, ..., \mathrm{V}\bar{n}, \mathrm{V}\overline{n+1}, \mathrm{W}1, ..., \mathrm{W}\bar{m}$
por variables $\mathrm{N}\overline{k_1}, ..., \mathrm{N}\overline{k_n}, \mathrm{N}\overline{k_{n+1}}, \mathrm{P}\overline{j_1}, ..., \mathrm{P}\overline{j_m}$
ninguna de las cuales es de las auxiliares antes seleccionadas, entonces la palabra obtenida es un programa que denotaremos con $\left[\mathrm{N}\overline{k_{n+1}} \leftarrow f(\mathrm{N}\overline{k_1}, ..., \mathrm{N}\overline{k_n}, \mathrm{P}\overline{j_1}, ..., \mathrm{P}\overline{j_m})\right]$
el cual debe tener la siguiente propiedad: - Si hacemos correr $\left[\mathrm{N}\overline{k_{n+1}} \leftarrow f(\mathrm{N}\overline{k_1}, ..., \mathrm{N}\overline{k_n}, \mathrm{P}\overline{j_1}, ..., \mathrm{P}\overline{j_m})\right]$ partiendo de cualquier estado que le asigne a las variables $\mathrm{N}\overline{k_1}, ..., \mathrm{N}\overline{k_n}, \mathrm{P}\overline{j_1}, ..., \mathrm{P}\overline{j_m}$ valores $x_1, ..., x_n, \alpha_1, ..., \alpha_m$, entonces $\left[\mathrm{N}\overline{k_{n+1}} \leftarrow f(\mathrm{N}\overline{k_1}, ..., \mathrm{N}\overline{k_n}, \mathrm{P}\overline{j_1}, ..., \mathrm{P}\overline{j_m})\right]$ termina si y solo si $(x_1, ..., x_n, \alpha_1, ..., \alpha$ D_f y en caso de terminacion se llega a un estado el cual le asigna a la variable $\mathrm{N}\overline{k_{n+1}}$ el valor $f(x_1, ..., x_n, \alpha_1, ..., \alpha_m)$ y a los contenidos de las variables $\mathrm{N}\overline{k_1}, ..., \mathrm{N}\overline{k_n}, \mathrm{P}\overline{j_1}, ..., \mathrm{P}\overline{j_m}$ no los modifica, salvo en el caso de que $\mathrm{N}\overline{k_i} = \mathrm{N}\overline{k_{n+1}}$, situacion en la cual el valor final de la variable $\mathrm{N}\overline{k_i}$ sera $f(x_1, ..., x_n, \alpha_1, ..., \alpha_m)$. El programa $\left[\mathrm{N}\overline{k_{n+1}} \leftarrow f(\mathrm{N}\overline{k_1}, ..., \mathrm{N}\overline{k_n}, \mathrm{P}\overline{j_1}, ..., \mathrm{P}\overline{j_m})\right]$ es comunmente llamado la expansion del macro con respecto a la eleccion de variables y labels realizada

Proposición 54

(a) Sea $f : S \subseteq \omega^n \times \Sigma^{*m} \to \omega$ una funcion Σ-computable. Entonces hay un macro $[\mathrm{V}\overline{n+1} \leftarrow f(\mathrm{V}1, ..., \mathrm{V}\bar{n}, \mathrm{W}1, ..., \mathrm{W}\bar{m})]$ (b) Sea $f : S \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ una funcion Σ-computable. Entonces hay un macro $[\mathrm{W}\overline{m+1} \leftarrow f(\mathrm{V}1, ..., \mathrm{V}\bar{n}, \mathrm{W}1, ..., \mathrm{W}\bar{m})]$ Prueba: (b) Sea $\mathcal{P}$ un programa que compute a f. Tomemos un k tal que $k \geq n, m$ y tal que todas las variables y labels de $\mathcal{P}$ estan en el conjunto
$\{\mathrm{N}1, ..., \mathrm{N}\bar{k}, \mathrm{P}1, ..., \mathrm{P}\bar{k}, \mathrm{L}1, ..., \mathrm{L}\bar{k}\}$.

Sea $\mathcal{P}'$ la palabra que resulta de reemplazar en $\mathcal{P}$: - la variable $\mathrm{N}\bar{j}$ por $\mathrm{V}\overline{n+j}$, para cada $j = 1, ..., k$ - la variable $\mathrm{P}\bar{j}$ por $\mathrm{W}\overline{m+j}$, para cada $j = 1, ..., k$ - el label $\mathrm{L}\bar{j}$ por $\mathrm{A}\bar{j}$, para cada $j = 1, ..., k$ Notese que

$$
\begin{array}{l}
\mathrm{V}\overline{n+1} \leftarrow \mathrm{V}1 \\
\quad\vdots \\
\mathrm{V}\overline{n+n} \leftarrow \mathrm{V}\bar{n} \\
\mathrm{V}\overline{n+n+1} \leftarrow 0 \\
\quad\vdots \\
\mathrm{V}\overline{n+k} \leftarrow 0 \\
\mathrm{W}\overline{m+1} \leftarrow \mathrm{W}1 \\
\quad\vdots \\
\mathrm{W}\overline{m+m} \leftarrow \mathrm{W}\bar{m} \\
\mathrm{W}\overline{m+m+1} \leftarrow \varepsilon \\
\quad\vdots \\
\mathrm{W}\overline{m+k} \leftarrow \varepsilon \\
\mathcal{P}'
\end{array}
$$

es el macro buscado, el cual tendra sus variables auxiliares y labels en la lista $\mathrm{V}\overline{n+1}, ..., \mathrm{V}\overline{n+k}, \mathrm{W}\overline{m+2}$

□

Proposición 55 Sea $P : S \subseteq \omega^n \times \Sigma^{*m} \to \omega$ un predicado Σ-computable. Entonces hay un macro $[\mathrm{IF}\ P(\mathrm{V}1, ..., \mathrm{V}\bar{n}, \mathrm{W}1, ..., \mathrm{W}\bar{m})\ \mathrm{GOTO}\ \mathrm{A}1]$ Usando macros podemos ahora probar el siguiente importante teorema.

Teorema 56 Si h es Σ-recursiva, entonces h es Σ -computable. Prueba: Probaremos por induccion en k que

(*) Si $h \in \mathrm{R}_k^\Sigma$, entonces h es Σ -computable. El caso $k = 0$ es dejado al lector. Supongamos (*) vale para k, veremos que vale para $k+1$. Sea $h \in \mathrm{R}_{k+1}^\Sigma - \mathrm{R}_k^\Sigma$. Hay varios casos

Caso 1. Supongamos $h = M(P)$, con $P : \omega \times \omega^n \times \Sigma^{*m} \to \omega$, un predicado perteneciente a R_k^Σ. Por hipotesis inductiva, P es Σ- computable y por lo tanto tenemos un macro

$[\mathrm{IF}\ P(\mathrm{V}1, ..., \mathrm{V}\overline{n+1}, \mathrm{W}1, ..., \mathrm{W}\bar{m})\ \mathrm{GOTO\ A}1]$

lo cual nos permite realizar el siguiente programa

$$\begin{array}{ll} \mathrm{L}2 & \mathrm{IF}\ P(\mathrm{N}\overline{n+1}, \mathrm{N}1, ..., \mathrm{N}\bar{n}, \mathrm{P}1, ..., \mathrm{P}\bar{m})\ \mathrm{GOTO\ L}1 \\ & \mathrm{N}\overline{n+1} \leftarrow \mathrm{N}\overline{n+1} + 1 \\ & \mathrm{GOTO\ L}2 \\ \mathrm{L}1 & \mathrm{N}1 \leftarrow \mathrm{N}\overline{n+1} \end{array}$$

Es facil chequear que este programa computa h. Caso 2. Supongamos $h = R(f, \mathcal{G})$, con

$$\begin{array}{lll} f & : & S_1 \times ... \times S_n \times L_1 \times ... \times L_m \to \Sigma^* \\ \mathcal{G}_a & : & S_1 \times ... \times S_n \times L_1 \times ... \times L_m \times \Sigma^* \times \Sigma^* \to \Sigma^*,\ a \in \Sigma \end{array}$$

elementos de R_k^Σ. Sea $\Sigma = \{a_1, ..., a_r\}$. Por hipotesis inductiva, las funciones f, $\mathcal{G}_a$, $a \in \Sigma$, son Σ-computables y por lo tanto podemos hacer el siguiente programa via el uso de macros

$$\begin{array}{ll} & [\mathrm{P}\overline{m+3} \leftarrow f(\mathrm{N}1, ..., \mathrm{N}\bar{n}, \mathrm{P}1, ..., \mathrm{P}\bar{m})] \\ \mathrm{L}\overline{r+1} & \mathrm{IF\ P}\overline{m+1}\ \ \mathrm{BEGINS}\ a_1\ \mathrm{GOTO\ L}1 \\ & \quad\vdots \\ & \mathrm{IF\ P}\overline{m+1}\ \mathrm{BEGINS}\ a_r\ \mathrm{GOTO\ L}\bar{r} \\ & \mathrm{GOTO\ L}\overline{r+2} \\ \mathrm{L}1 & \mathrm{P}\overline{m+1} \leftarrow\ {}^\frown\mathrm{P}\overline{m+1} \\ & [\mathrm{P}\overline{m+3} \leftarrow \mathcal{G}_{a_1}(\mathrm{N}1, ..., \mathrm{N}\bar{n}, \mathrm{P}1, ..., \mathrm{P}\bar{m}, \mathrm{P}\overline{m+2}, \mathrm{P}\overline{m+3})] \\ & \mathrm{P}\overline{m+2} \leftarrow \mathrm{P}\overline{m+2}a_1 \\ & \mathrm{GOTO\ L}\overline{r+1} \\ & \quad\vdots \\ \mathrm{L}\bar{r} & \mathrm{P}\overline{m+1} \leftarrow\ {}^\frown\mathrm{P}\overline{m+1} \\ & \mathrm{P}\overline{m+3} \leftarrow \mathcal{G}_{a_r}(\mathrm{N}1, ..., \mathrm{N}\bar{n}, \mathrm{P}1, ..., \mathrm{P}\bar{m}, \mathrm{P}\overline{m+2}, \mathrm{P}\overline{m+3}) \\ & \mathrm{P}\overline{m+2} \leftarrow \mathrm{P}\overline{m+2}a_r \\ & \mathrm{GOTO\ L}\overline{r+1} \\ \mathrm{L}\overline{r+2} & \mathrm{P}1 \leftarrow \mathrm{P}\overline{m+3} \end{array}$$

Es facil chequear que este programa computa h. El resto de los casos son dejados al lector.
$\square$

Analisis de la recursividad de $\mathcal{S}^\Sigma$

Primero probaremos dos lemas que muestran que la sintaxis de $\mathcal{S}^\Sigma$ es $(\Sigma \cup \Sigma_p)$-recursiva primitiva. Recordemos que $S : Num^* \to Num^*$ fue definida de la siguiente manera

$$\begin{array}{rcl} S(\varepsilon) & = & 1 \\ S(\alpha 0) & = & \alpha 1 \\ S(\alpha 1) & = & \alpha 2 \\ S(\alpha 2) & = & \alpha 3 \\ S(\alpha 3) & = & \alpha 4 \\ S(\alpha 4) & = & \alpha 5 \\ S(\alpha 5) & = & \alpha 6 \\ S(\alpha 6) & = & \alpha 7 \\ S(\alpha 7) & = & \alpha 8 \\ S(\alpha 8) & = & \alpha 9 \\ S(\alpha 9) & = & S(\alpha)0 \end{array}$$

Tambien $\bar{\ } : \omega \to Num^*$ fue definida de la siguiente manera

$$\begin{array}{rcl} \bar{0} & = & \varepsilon \\ \overline{n+1} & = & S(\bar{n}) \end{array}$$

Es obvio de las definiciones que ambas funciones son Num-p.r.. Mas aun tenemos

Lema 57 Sea Σ un alfabeto cualquiera. Las funciones S y $\_$ son $(\Sigma \cup \Sigma_p)$-p.r.. Prueba: Use el Teorema 51. $\square$

Recordemos que $Bas : \text{Ins}^\Sigma \rightarrow (\Sigma \cup \Sigma_p)^*$, fue definida por

$$Bas(I) = \begin{cases} J & \text{si } I \text{ es de la forma L}\bar{k}J \text{ con } J \in \text{Ins}^\Sigma \\ I & \text{caso contrario} \end{cases}$$

Definamos $Lab : \text{Ins}^\Sigma \rightarrow (\Sigma \cup \Sigma_p)^*$ de la siguiente manera $Lab(I) = \begin{cases} \text{L}\bar{k} & \text{si } I \text{ es de la forma L}\bar{k}J \text{ con} \\ \varepsilon & \text{caso contrario} \end{cases}$

Lema 58 Para cada $n, x \in \omega$, tenemos que $|\bar{n}| \leq x$ si y solo si $n \leq 10^x - 1$

Lema 59 Ins^Σ es un conjunto $(\Sigma \cup \Sigma_p)$-p.r.. Prueba: Para simplificar la prueba asumiremos que $\Sigma = \{@, \&\}$. Ya que Ins^Σ es union de los siguientes conjuntos

$$\begin{array}{rcl}
L_1 & = & \left\{\text{N}\bar{k} \leftarrow \text{N}\bar{k} + 1 : k \in \mathbf{N}\right\} \\
L_2 & = & \left\{\text{N}\bar{k} \leftarrow \text{N}\bar{k} \dot{-} 1 : k \in \mathbf{N}\right\} \\
L_3 & = & \left\{\text{N}\bar{k} \leftarrow \text{N}\bar{n} : k, n \in \mathbf{N}\right\} \\
L_4 & = & \left\{\text{N}\bar{k} \leftarrow 0 : k \in \mathbf{N}\right\} \\
L_5 & = & \left\{\text{IF N}\bar{k} \neq 0 \text{ GOTO L}\bar{m} : k, m \in \mathbf{N}\right\} \\
L_6 & = & \left\{\text{P}\bar{k} \leftarrow \text{P}\bar{k}.@ : k \in \mathbf{N}\right\} \\
L_7 & = & \left\{\text{P}\bar{k} \leftarrow \text{P}\bar{k}.\& : k \in \mathbf{N}\right\} \\
L_8 & = & \left\{\text{P}\bar{k} \leftarrow {}^\frown\text{P}\bar{k} : k \in \mathbf{N}\right\} \\
L_9 & = & \left\{\text{P}\bar{k} \leftarrow \text{P}\bar{n} : k, n \in \mathbf{N}\right\} \\
L_9 & = & \left\{\text{P}\bar{k} \leftarrow \varepsilon : k \in \mathbf{N}\right\} \\
L_{10} & = & \left\{\text{IF P}\bar{k} \text{ BEGINS @ GOTO L}\bar{m} : k, m \in \mathbf{N}\right\} \\
L_{11} & = & \left\{\text{IF P}\bar{k} \text{ BEGINS \& GOTO L}\bar{m} : k, m \in \mathbf{N}\right\} \\
L_{12} & = & \{\text{GOTO L}\bar{m} : m \in \mathbf{N}\} \\
L_{13} & = & \{\text{SKIP}\} \\
L_{14} & = & \left\{\text{L}\bar{k}\alpha : k \in \mathbf{N} \text{ y } \alpha \in L_1 \cup ... \cup L_{13}\right\}
\end{array}$$

solo debemos probar que $L_1, ..., L_{14}$ son $(\Sigma \cup \Sigma_p)$ -p.r.. Veremos primero por ejemplo que $L_{10} = \left\{\text{IFP}\bar{k}\text{BEGINS@GOTOL}\bar{m} : k, m \in \mathbf{N}\right\}$

es $(\Sigma \cup \Sigma_p)$-p.r.. Primero notese que $\alpha \in L_{10}$ si y solo si existen $k, m \in \mathbf{N}$ tales que $\alpha = \text{IFP}\bar{k}\text{BEGINS@GOTOL}\bar{m}$

Mas formalmente tenemos que $\alpha \in L_{10}$ si y solo si $(\exists k \in \mathbf{N})(\exists m \in \mathbf{N})\ \alpha = \text{IFP}\bar{k}\text{BEGINS@GOTOL}\bar{m}$

Ya que cuando existen tales k, m tenemos que $\bar{k}$ y $\bar{m}$ son subpalabras de α, el lema anterior nos dice que $\alpha \in L_{10}$ si y solo si $(\exists k \in \mathbf{N})_{k \leq 10^{|\alpha|}}(\exists m \in \mathbf{N})_{m \leq 10^{|\alpha|}}\ \alpha = \text{IFP}\bar{k}\text{BEGINS@GOTOL}\bar{m}$

Sea $P = \lambda mk\alpha \left[\alpha = \text{IFP}\bar{k}\text{BEGINS@GOTOL}\bar{m}\right]$

Ya que $D_{\lambda k\left[\bar{k}\right]} = \omega$, tenemos que $D_P = \omega \times (\Sigma \cup \Sigma_p)^* \times (\Sigma \cup \Sigma_p)^*$. Notese que $P = \lambda\alpha\beta\left[\alpha = \beta\right] \circ \left(p_3^{2,1}, f\right)$

donde $f = \lambda\alpha_1\alpha_2\alpha_3\alpha_4\left[\alpha_1\alpha_2\alpha_3\alpha_4\right] \circ \left(C^{2,1}_{\text{IFP}}, \lambda k\left[\bar{k}\right] \circ p_2^{2,1}, C^{2,1}_{\text{BEGINS@GOTOL}}, \lambda k\left[\bar{k}\right] \circ p_1^{2,1}\right)$

lo cual nos dice que P es $(\Sigma \cup \Sigma_p)$-p.r.. Notese que

$\chi_{L_{10}} = \lambda\alpha\left[(\exists k \in \mathbf{N})_{k \leq 10^{|\alpha|}}(\exists m \in \mathbf{N})_{m \leq 10^{|\alpha|}}\ P(m, k, \alpha)\right]$

Esto nos dice que podemos usar dos veces el Lema 39 para ver que $\chi_{L_{10}}$ es $(\Sigma \cup \Sigma_p)$-p.r.. Veamos como. Sea $Q = \lambda k\alpha\left[(\exists m \in \mathbf{N})_{m \leq 10^{|\alpha|}}\ P(m, k, \alpha)\right]$

Por el Lema 39 tenemos que $\lambda xk\alpha\left[(\exists m \in \mathbf{N})_{m \leq x}\ P(m, k, \alpha)\right]$

es $(\Sigma \cup \Sigma_p)$-p.r. lo cual nos dice que $Q = \lambda xk\alpha\left[(\exists m \in \mathbf{N})_{m \leq x}\ P(m, k, \alpha)\right] \circ (\lambda\alpha\left[10^{|\alpha|}\right] \circ p_2^{1,1}, p_1^{1,1}, p_2^{1,1})$

lo es. Ya que $\chi_{L_{10}} = \lambda\alpha\left[(\exists k \in \mathbf{N})_{k \leq 10^{|\alpha|}}\ Q(k, \alpha)\right]$

podemos en forma similar aplicar el Lema 39 y obtener finalmente que $\chi_{L_{10}}$ es $(\Sigma\cup\Sigma_p)$-p.r.. En forma similar podemos probar que $L_1,...,L_{13}$ son $(\Sigma\cup\Sigma_p)$-p.r.. Esto nos dice que $L_1\cup...\cup L_{13}$ es $(\Sigma\cup\Sigma_p)$-p.r.. Notese que $L_1\cup...\cup L_{13}$ es el conjunto de las instrucciones basicas de $\mathcal{S}^\Sigma$. Llamemos InsBas^Σ a dicho conjunto. Para ver que L_{14} es $(\Sigma\cup\Sigma_p)$-p.r. notemos que

$\chi_{L_{14}} = \lambda\alpha\left[(\exists k\in\mathbf{N})_{k\leq 10^{|\alpha|}}(\exists\beta\in\mathrm{InsBas}^\Sigma)_{|\beta|\leq|\alpha|}\ \alpha = \mathrm{L}\bar{k}\beta\right]$

lo cual nos dice que aplicando dos veces el Lema 39 obtenemos que $\chi_{L_{14}}$ es $(\Sigma\cup\Sigma_p)$-p.r.. Ya que $\mathrm{Ins}^\Sigma = \mathrm{InsBas}^\Sigma\cup L_{14}$ tenemos que Ins^Σ es $(\Sigma\cup\Sigma_p)$-p.r.. $\square$

Lema 60 *Bas* y *Lab* son funciones $(\Sigma \cup \Sigma_p)$-p.r. Prueba: Sea $<$ un orden total estricto sobre $\Sigma \cup \Sigma_p$. Sea $L = \{\mathrm{L}\bar{k} : k \in \mathbf{N}\} \cup \{\varepsilon\}$. Dejamos al lector probar que L es un conjunto $(\Sigma \cup \Sigma_p)$-p.r.. Sea

$P = \lambda I\alpha \left[\alpha \in \mathrm{Ins}^\Sigma \wedge I \in \mathrm{Ins}^\Sigma \wedge [\alpha]_1 \neq \mathrm{L} \wedge (\exists \beta \in L)\ I = \beta\alpha\right]$

Note que $D_P = (\Sigma \cup \Sigma_p)^{*2}$. Dejamos al lector probar que P es $(\Sigma \cup \Sigma_p)$-p.r.. Notese ademas que cuando $I \in \mathrm{Ins}^\Sigma$ tenemos que $P(I, \alpha) = 1$ sii $\alpha = Bas(I)$. Dejamos al lector probar que $Bas = M^<(P)$ por lo que para ver que Bas es $(\Sigma \cup \Sigma_p)$-p.r., solo nos falta ver que la funcion Bas es acotada por alguna funcion $(\Sigma \cup \Sigma_p)$-p.r. y $(\Sigma \cup \Sigma_p)$-total. Pero esto es trivial ya que $|Bas(I)| \leq |I| = p_1^{0,1}(I)$ para cada $I \in \mathrm{Ins}^\Sigma$. Finalmente note que

$Lab = M^<(\lambda I\alpha\,[\alpha Bas(I) = I])$

lo cual nos dice que Lab es $(\Sigma \cup \Sigma_p)$-p.r.. $\square$ Recordemos que dado un programa $\mathcal{P}$ habiamos definido $I_i^{\mathcal{P}} = \varepsilon$, para $i = 0$ o $i > n(\mathcal{P})$. O sea que la funcion $(\Sigma \cup \Sigma_p)$-mixta $\lambda i\mathcal{P}\left[I_i^{\mathcal{P}}\right]$ tiene dominio igual a $\omega \times \mathrm{Pro}^\Sigma$.

Lema 61 (a) Pro^Σ es un conjunto $(\Sigma \cup \Sigma_p)$-p.r. (b) $\lambda\mathcal{P}\,[n(\mathcal{P})]$ y $\lambda i\mathcal{P}\left[I_i^{\mathcal{P}}\right]$ son funciones $(\Sigma \cup \Sigma_p)$-p.r.. Prueba: Ya que $\mathrm{Pro}^\Sigma = D_{\lambda\mathcal{P}[n(\mathcal{P})]}$ tenemos que (b) implica (a). Para probar (b) sea $<$ un orden total estricto sobre $\Sigma \cup \Sigma_p$. Sea P el siguiente predicado

$\lambda x \Big[Lt(x) > 0 \wedge (\forall t \in \mathbf{N})_{t \leq Lt(x)}\ *^<((x)_t) \in \mathrm{Ins}^\Sigma \wedge$ $(\forall t \in \mathbf{N})_{t \leq Lt(x)} (\forall m \in \mathbf{N})\ \neg(\mathrm{L}\bar{m}$ t-final $*^<((x)_t)) \vee$ $(\exists j \in \mathbf{N})_{j \leq Lt(x)} (\exists \alpha \in (\Sigma \cup \Sigma_p) - Num)\ \mathrm{L}\bar{m}\alpha$ t-

Notese que $D_P = \mathbf{N}$ y que $P(x) = 1$ sii $Lt(x) > 0$, $*^<((x)_t) \in \mathrm{Ins}^\Sigma$, para cada $t = 1, ..., Lt(x)$ y ademas $\subset_{t=1}^{t=Lt(x)} *^<((x)_t) \in \mathrm{Pro}^\Sigma$. Para ver que P es $(\Sigma \cup \Sigma_p)$-p.r. solo nos falta acotar el cuantificador $(\forall m \in \mathbf{N})$ de la expresion lambda que define a P. Ya que nos interesan los valores de m para los cuales $\bar{m}$ es posiblemente una subpalabra de alguna de las palabras $*^<((x)_j)$, el Lema 58 nos dice que una cota posible es $10^{\max\{|*^<((x)_j)| : 1 \leq j \leq Lt(x)\}} - 1$. Dejamos al lector los detalles de la prueba de que P es $(\Sigma \cup \Sigma_p)$-p.r.. Sea

$Q = \lambda x\alpha \left[P(x) \wedge \alpha = \subset_{t=1}^{t=Lt(x)} *^<((x)_t)\right]$.

Note que $D_Q = \mathbf{N} \times (\Sigma \cup \Sigma_p)^*$. Claramente Q es $(\Sigma \cup \Sigma_p)$-p.r.. Ademas note que $D_{M(Q)} = \mathrm{Pro}^\Sigma$. Notese que para $\mathcal{P} \in \mathrm{Pro}^\Sigma$, tenemos que $M(Q)(\mathcal{P})$ es aquel numero tal que pensado como infinitupla (via mirar su secuencia de exponentes) codifica la secuencia de instrucciones que forman a $\mathcal{P}$. Es decir $M(Q)(\mathcal{P}) = \left\langle \#^<(I_1^{\mathcal{P}}), \#^<(I_2^{\mathcal{P}}), ..., \#^<(I_{n(\mathcal{P})}^{\mathcal{P}}), 0, 0, ... \right\rangle$

Por (b) del Lema 43, $M(Q)$ es $(\Sigma \cup \Sigma_p)$ -p.r. ya que para cada $\mathcal{P} \in \mathrm{Pro}^\Sigma$ tenemos que

$$\begin{array}{rcl} M(Q)(\mathcal{P}) & = & \left\langle \#^<(I_1^{\mathcal{P}}), \#^<(I_2^{\mathcal{P}}), ..., \#^<(I_{n(\mathcal{P})}^{\mathcal{P}}), 0, 0, ... \right\rangle \\ & = & \prod\limits_{i=1}^{n(\mathcal{P})} pr(i)^{\#^<(I_1^{\mathcal{P}})} \\ & \leq & \prod\limits_{i=1}^{|\mathcal{P}|} pr(i)^{\#^<(\mathcal{P})} \end{array}$$

Ademas tenemos que $\begin{array}{rcl} \lambda\mathcal{P}\,[n(\mathcal{P})] & = & \lambda x\,[Lt(x)] \circ M(Q) \\ \lambda i\mathcal{P}\left[I_i^{\mathcal{P}}\right] & = & *^< \circ g \circ \left(p_1^{1,1}, M(Q) \circ p_2^{1,1}\right) \end{array}$

donde $g = C_0^{1,1}\ |_{\{0\} \times \omega} \cup \lambda ix\,[(x)_i]$, lo cual dice que $\lambda\mathcal{P}\,[n(\mathcal{P})]$ y $\lambda i\mathcal{P}\left[I_i^{\mathcal{P}}\right]$ son funciones $(\Sigma \cup \Sigma_p)$-p.r.. $\square$

Las funciones $i^{n,m}$, $E_\#^{n,m}$ y $E_*^{n,m}$

Sean $n, m \geq 0$ fijos. Definamos entonces las funciones

$$\begin{array}{rcl} i^{n,m} & : & \omega \times \omega^n \times \Sigma^{*m} \times \mathrm{Pro}^\Sigma \to \omega \\ E_\#^{n,m} & : & \omega \times \omega^n \times \Sigma^{*m} \times \mathrm{Pro}^\Sigma \to \omega^{[\mathbf{N}]} \\ E_*^{n,m} & : & \omega \times \omega^n \times \Sigma^{*m} \times \mathrm{Pro}^\Sigma \to \Sigma^{*[\mathbf{N}]} \end{array}$$

de la siguiente manera $(i^{n,m}(0, \vec{x}, \vec{\alpha}, \mathcal{P}), E_\#^{n,m}(0, \vec{x}, \vec{\alpha}, \mathcal{P}), E_*^{n,m}(0, \vec{x}, \vec{\alpha}, \mathcal{P})) =$

$= (1, (x_1, ..., x_n, 0, 0, ...), (\alpha_1, ..., \alpha_m, \varepsilon, \varepsilon, ...))$

$(i^{n,m}(t+1,\vec{x},\vec{\alpha},\mathcal{P}),E_{\#}^{n,m}(t+1,\vec{x},\vec{\alpha},\mathcal{P}),E_{*}^{n,m}(t+1,\vec{x},\vec{\alpha},\mathcal{P}))=$
$=DIS_{\mathcal{P}}(i^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),E_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),E_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}))$

Notese que

$(i^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),E_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),E_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}))$

es la descripcion instantanea que se obtiene luego de correr $\mathcal{P}$ una cantidad t de pasos a partir de la descripcion instantanea $(1,(x_1,...,x_n,0,0,...),(\alpha_1,...,\alpha_m,0,0,...))$. Es importante notar que si bien $i^{n,m}$ es una funcion $(\Sigma\cup\Sigma_p)$-mixta, ni $E_{\#}^{n,m}$ ni $E_{*}^{n,m}$ lo son. Definamos para cada $j\in\mathbf{N}$, funciones

$$\begin{array}{rcl} E_{\#j}^{n,m} & : & \omega\times\omega^n\times\Sigma^{*m}\times\mathrm{Pro}^{\Sigma}\to\omega \\ E_{*j}^{n,m} & : & \omega\times\omega^n\times\Sigma^{*m}\times\mathrm{Pro}^{\Sigma}\to\Sigma^* \end{array}$$

de la siguiente manera $\begin{array}{rcl} E_{\#j}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}) & = & j\text{-esima coordenada de } E_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}) \\ E_{*j}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}) & = & j\text{-esima coordenada de } E_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}) \end{array}$

Notese que $\begin{array}{rcl} E_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}) & = & (E_{\#1}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),E_{\#2}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),...) \\ E_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}) & = & (E_{*1}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),E_{*2}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),...) \end{array}$

Nuestro proximo objetivo es mostrar que las funciones $i^{n,m}$, $E_{\#j}^{n,m}$, $E_{*j}^{n,m}$ son $(\Sigma\cup\Sigma_p)$-p.r. Para esto primero debemos probar un lema el cual muestre que una ves codificadas las descripciones instantaneas en forma numerica, las funciones que dan la descripcion instantanea sucesora son $(\Sigma\cup\Sigma_p)$ -p.r.. Dado un orden total estricto $<$ sobre $\Sigma\cup\Sigma_p$, codificaremos las descripciones instantaneas haciendo uso de las biyecciones

$$\begin{array}{rcl} \omega^{[\mathbf{N}]} & \to & \mathbf{N} \\ (s_1,s_2,...) & \to & \langle s_1,s_2,...\rangle \end{array} \qquad \begin{array}{rcl} \Sigma^{*[\mathbf{N}]} & \to & \mathbf{N} \\ (\sigma_1,\sigma_2,...) & \to & \langle \#^<(\sigma_1),\#^<(\sigma_2),...\rangle \end{array}$$

Es decir que a la descripcion instantanea $(i,(s_1,s_2,...),(\sigma_1,\sigma_2,...))$

la codificaremos con la terna $(i,\langle s_1,s_2,...\rangle,\langle \#^<(\sigma_1),\#^<(\sigma_2),...\rangle)\in\omega\times\mathbf{N}\times\mathbf{N}$

Es decir que una terna $(i,x,y)\in\omega\times\mathbf{N}\times\mathbf{N}$ codificara a la descripcion instantanea $(i,((x)_1,(x)_2,...),(*^<((y)_1),*^<((y)_2),...))$

Definamos $\begin{array}{rcl} s & : & \omega\times\mathbf{N}\times\mathbf{N}\times\mathrm{Pro}^{\Sigma}\to\omega \\ S_{\#} & : & \omega\times\mathbf{N}\times\mathbf{N}\times\mathrm{Pro}^{\Sigma}\to\omega \\ S_{*} & : & \omega\times\mathbf{N}\times\mathbf{N}\times\mathrm{Pro}^{\Sigma}\to\omega \end{array}$

de la siguiente manera $s(i,x,y,\mathcal{P})=$ primera coordenada de la codificacion de la descripcion instanta
sucesora de $(i,((x)_1,(x)_2,...),(*^<((y)_1),*^<((y)_2),...))$ en $\mathcal{P}$

$S_{\#}(i,x,y,\mathcal{P})=$ segunda coordenada de la codificacion de la descripcion instantanea
sucesora de $(i,((x)_1,(x)_2,...),(*^<((y)_1),*^<((y)_2),...))$ en $\mathcal{P}$

$S_{*}(i,x,y,\mathcal{P})=$ tercera coordenada de la codificacion de la descripcion instantanea
sucesora de $(i,((x)_1,(x)_2,...),(*^<((y)_1),*^<((y)_2),...))$ en $\mathcal{P}$

Notese que la definicion de estas funciones depende del orden total estricto $<$ sobre $\Sigma\cup\Sigma_p$.

Lema 62 Dado un orden total estricto $<$ sobre $\Sigma\cup\Sigma_p$, las funciones s, $S_{\#}$ y S_* son $(\Sigma\cup\Sigma_p)$-p.r.. Prueba: Necesitaremos algunas funciones $(\Sigma\cup\Sigma_p)$-p.r.. Dada una instruccion I en la cual al menos ocurre una variable, usaremos $\#Var1(I)$ para denotar el numero de la primer variable que ocurre en I. Por ejemplo

$\#Var1\left(\text{L}\bar{n}\text{ IF N}\bar{k}\neq 0\text{ GOTO L}\bar{m}\right)=k$

Notese que $\lambda I[\#Var1(I)]$ tiene dominio igual a $\text{Ins}^{\Sigma}-L$, donde L es la union de los siguientes conjuntos

$$\{\text{GOTOL}\bar{m}:m\in\mathbf{N}\}\cup\{\text{L}\bar{k}\text{GOTOL}\bar{m}:k,m\in\mathbf{N}\}\ \{\text{SKIP}\}\cup\{\text{L}\bar{k}\text{SKIP}:k\in\mathbf{N}\}$$

Dada una instruccion I en la cual ocurren dos variables, usaremos $\#Var2(I)$ para denotar el numero de la segunda variable que ocurre en I. Por ejemplo $\#Var2\left(\text{N}\bar{k}\leftarrow\text{N}\bar{m}\right)=m$

Notese que el dominio de $\lambda I[\#Var2(I)]$ es igual a la union de los siguientes conjuntos
$\{\text{N}\bar{k}\ \leftarrow\ \text{N}\bar{m}:k,m\in\mathbf{N}\}\cup\{\text{L}\bar{j}\ \text{N}\bar{k}\leftarrow\text{N}\bar{m}:j,k,m\in\mathbf{N}\}$
$\{\text{P}\bar{k}\ \leftarrow\ \text{P}\bar{m}:k,m\in\mathbf{N}\}\cup\{\text{L}\bar{j}\ \text{P}\bar{k}\leftarrow\text{P}\bar{m}:j,k,m\in\mathbf{N}\}$

Ademas notese que para una instruccion I tenemos que $\begin{array}{rcl}\#Var1(I) & = & \min_k(\text{N}\bar{k}\leftarrow\text{ ocu }I\vee\text{N}\bar{k}\neq\text{ ocu }I\\ \#Var2(I) & = & \min_k(\text{N}\bar{k}\text{ t-final }I\vee\text{N}\bar{k}+\text{ ocu }I\end{array}$

Esto nos dice que si llamamos P al predicado $\lambda k\alpha\left[\alpha\in\text{Ins}^{\Sigma}\wedge(\text{N}\bar{k}\leftarrow\text{ ocu }\alpha\vee\text{N}\bar{k}\neq\text{ ocu }\alpha\vee\text{P}\bar{k}\leftarrow\text{ ocu }\alpha\right.$
entonces $\lambda I[\#Var1(I)]=M(P)$ por lo cual $\lambda I[\#Var1(I)]$ es $(\Sigma\cup\Sigma_p)$-p.r. Similarmente se puede ver que $\lambda I[\#Var2(I)]$ es $(\Sigma\cup\Sigma_p)$-p.r.. Sea $\begin{array}{rcl}F_{\dot{-}}:\mathbf{N}\times\mathbf{N} & \rightarrow & \omega\\ (x,j) & \rightarrow & \langle(x)_1,....,(x)_{j-1},(x)_j\dot{-}1,(x)_{j+1},...\rangle\end{array}$

Ya que $F_{\dot{-}}(x,j)=\begin{cases}Q(x,pr(j)) & \text{si }pr(j)\text{ divide }x\\ x & \text{caso contrario}\end{cases}$

tenemos que $F_{\dot{-}}$ es $(\Sigma\cup\Sigma_p)$-p.r.. Sea $\begin{array}{rcl}F_+:\mathbf{N}\times\mathbf{N} & \rightarrow & \omega\\ (x,j) & \rightarrow & \langle(x)_1,....,(x)_{j-1},(x)_j+1,(x)_{j+1},...\rangle\end{array}$

Ya que $F_+(x,j)=x.pr(j)$ tenemos que F_+ es $(\Sigma\cup\Sigma_p)$-p.r.. Sea $\begin{array}{rcl}F_{\leftarrow}:\mathbf{N}\times\mathbf{N}\times\mathbf{N} & \rightarrow & \omega\\ (x,j,k) & \rightarrow & \langle(x)_1,....,(x)_{j-1},\end{array}$

Ya que $F_{\leftarrow}(x,j,k)=Q(x,pr(j)^{(x)_j}).pr(j)^{(x)_k}$ tenemos que $F_{\leftarrow}$ es $(\Sigma\cup\Sigma_p)$-p.r.. Sea $\begin{array}{rcl}F_0:\mathbf{N}\times\mathbf{N} & \rightarrow & \omega\\ (x,j) & \rightarrow & \langle(x\end{array}$

Es facil ver que F_0 es $(\Sigma\cup\Sigma_p)$-p.r.. Para cada $a\in\Sigma$, sea $\begin{array}{rcl}F_a:\mathbf{N}\times\mathbf{N} & \rightarrow & \omega\\ (x,j) & \rightarrow & \langle(x)_1,....,(x)_{j-1},\#^{<}(*^{<}((x)_j)\end{array}$

Es facil ver que F_a es $(\Sigma\cup\Sigma_p)$-p.r.. En forma similar puede ser probado que $\begin{array}{rcl}F_{\frown}:\mathbf{N}\times\mathbf{N} & \rightarrow & \omega\\ (x,j) & \rightarrow & \langle(x)_1,....,\end{array}$

es $(\Sigma\cup\Sigma_p)$-p.r. Dado $(i,x,y,\mathcal{P})\in\omega\times\mathbf{N}\times\mathbf{N}\times\text{Pro}^{\Sigma}$, tenemos varios casos en los cuales los valores $s(i,x,y,\mathcal{P}),S_{\#}(i,x,y,\mathcal{P})$ y $S_*(i,x,y,\mathcal{P})$ pueden ser obtenidos usando las funciones antes definidas:

(1) CASO $i=0\vee i>n(\mathcal{P})$. Entonces $\begin{array}{rcl}s(i,x,y,\mathcal{P}) & = & i\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_*(i,x,y,\mathcal{P}) & = & y\end{array}$

(2) CASO $(\exists j\in\omega)\ Bas(I_i^{\mathcal{P}})=\text{N}\bar{j}\leftarrow\text{N}\bar{j}+1$. Entonces $\begin{array}{rcl}s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & F_+(x,\#Var1(I_i^{\mathcal{P}}))\\ S_*(i,x,y,\mathcal{P}) & = & y\end{array}$

(3) CASO $(\exists j\in\omega)\ Bas(I_i^{\mathcal{P}})=\text{N}\bar{j}\leftarrow\text{N}\bar{j}\dot{-}1$. Entonces $\begin{array}{rcl}s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & F_{\dot{-}}(x,\#Var1(I_i^{\mathcal{P}}))\\ S_*(i,x,y,\mathcal{P}) & = & y\end{array}$

(4) CASO $(\exists j,k\in\omega)\ Bas(I_i^{\mathcal{P}})=\text{N}\bar{j}\leftarrow\text{N}\bar{k}$. Entonces $\begin{array}{rcl}s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & F_{\leftarrow}(x,\#Var1(I_i^{\mathcal{P}}),\#Var2(I_i^{\mathcal{P}}\\ S_*(i,x,y,\mathcal{P}) & = & y\end{array}$

(5) CASO $(\exists j,k\in\omega)$ $Bas(I_i^{\mathcal{P}})=\mathrm{N}\bar{j}\leftarrow 0$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & F_0(x,\#Var1(I_i^{\mathcal{P}}))\\ S_{*}(i,x,y,\mathcal{P}) & = & y \end{array}$

(6) CASO $(\exists j,m\in\omega)$ $\left(Bas(I_i^{\mathcal{P}})=\mathrm{IF\ N}\bar{j}\neq 0\ \mathrm{GOTO\ L}\bar{m}\wedge (x)_j=0\right)$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & i+\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & y \end{array}$

(7) CASO $(\exists j,m\in\omega)$ $\left(Bas(I_i^{\mathcal{P}})=\mathrm{IF\ N}\bar{j}\neq 0\ \mathrm{GOTO\ L}\bar{m}\wedge (x)_j\neq 0\right)$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & \text{mí}\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & y \end{array}$

(8) CASO $(\exists j\in\omega)$ $Bas(I_i^{\mathcal{P}})=\mathrm{P}\bar{j}\leftarrow \mathrm{P}\bar{j}.a$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & F_a(y,\#Var1(I_i^{\mathcal{P}})) \end{array}$

(9) CASO $(\exists j\in\omega)$ $Bas(I_i^{\mathcal{P}})=\mathrm{P}\bar{j}\leftarrow {}^{\frown}\mathrm{P}\bar{j}$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & F_{\frown}(y,\#Var1(I_i^{\mathcal{P}})) \end{array}$

(10) CASO $(\exists j,k\in\omega)$ $Bas(I_i^{\mathcal{P}})=\mathrm{P}\bar{j}\leftarrow \mathrm{P}\bar{k}$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & F_{\leftarrow}(y,\#Var1(I_i^{\mathcal{P}}),\#Var2(I \end{array}$

(11) CASO $(\exists j\in\omega)$ $Bas(I_i^{\mathcal{P}})=\mathrm{P}\bar{j}\leftarrow \varepsilon$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & F_0(y,\#Var1(I_i^{\mathcal{P}})) \end{array}$

(12) CASO $(\exists j,m\in\omega)(\exists a\in\Sigma)$ $\left(Bas(I_i^{\mathcal{P}})=\mathrm{IF\ P}\bar{j}\ \mathrm{BEGINS}\ a\ \mathrm{GOTO\ L}\bar{m}\wedge [*^{<}((y)_j)]_1\neq a\right)$.

Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & i+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & y \end{array}$

(13) CASO $(\exists j,m\in\omega)(\exists a\in\Sigma)$ $\left(Bas(I_i^{\mathcal{P}})=\mathrm{IF\ P}\bar{j}\ \mathrm{BEGINS}\ a\ \mathrm{GOTO\ L}\bar{m}\wedge [*^{<}((y)_j)]_1= a\right)$.

Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & \text{mín}_l\left(Lab(I_l^{\mathcal{P}})\neq\varepsilon\wedge Lab(I_l^{\mathcal{P}})\text{ t -final }I_i^{\mathcal{P}}\right)\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & y \end{array}$

(14) CASO $(\exists j\in\omega)$ $Bas(I_i^{\mathcal{P}})=\mathrm{GOTO\ L}\bar{j}$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & \text{mín}_l\left(Lab(I_l^{\mathcal{P}})\neq\varepsilon\wedge Lab(I_l^{\mathcal{P}})\right.\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & y \end{array}$

(15) CASO $Bas(I_i^{\mathcal{P}})=\mathrm{SKIP}$. Entonces $\begin{array}{rcl} s(i,x,y,\mathcal{P}) & = & k+1\\ S_{\#}(i,x,y,\mathcal{P}) & = & x\\ S_{*}(i,x,y,\mathcal{P}) & = & y \end{array}$

O sea que los casos anteriores nos permiten definir conjuntos $S_1,...,S_{15}$, los cuales son disjuntos de a pares y cuya union da el conjunto $\omega\times\mathbf{N}\times\mathbf{N}\times\mathrm{Pro}^{\Sigma}$, de manera que cada una de las funciones $s,S_{\#}$ y S_* pueden escribirse como union disjunta de funciones $(\Sigma\cup\Sigma_p)$-p.r. restrinjidas respectivamente a los conjuntos $S_1,...,S_{15}$. Ya que los conjuntos $S_1,...,S_{15}$ son $(\Sigma\cup\Sigma_p)$ -p.r. el Lema 35 nos dice que $s,S_{\#}$ y S_* lo son. $\Box$

Aparte del lema anterior, para probar que las funciones $i^{n,m}$, $E_{\#}^{n,m}$ y $E_{*}^{n,m}$ son $(\Sigma\cup\Sigma_p)$-p.r., nos sera necesario el siguiente resultado. Recordemos que para $x_1,...,x_n\in\omega$, usabamos $\langle x_1,...,x_n\rangle$ para denotar $\langle x_1,...,x_n,0,...\rangle$. Ademas recordemos que en el Lema 62. Definamos

$$\begin{array}{rcl} C_{\#}^{n,m} & = & \lambda t\vec{x}\vec{\alpha}\mathcal{P}\left[\left\langle E_{\#1}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),E_{\#2}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),...\right\rangle\right]\\ C_{*}^{n,m} & = & \lambda t\vec{x}\vec{\alpha}\mathcal{P}\left[\langle \#^{<}(E_{*1}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P})),\#^{<}(E_{*2}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P})),...\rangle\right] \end{array}$$

Notese que
$$\begin{array}{rcl} i^{n,m}(0,\vec{x},\vec{\alpha},\mathcal{P}) & = & 1 \\ C_{\#}^{n,m}(0,\vec{x},\vec{\alpha},\mathcal{P}) & = & \langle x_1,...,x_n\rangle \\ C_{*}^{n,m}(0,\vec{x},\vec{\alpha},\mathcal{P}) & = & \langle \#^{<}(\alpha_1),...,\#^{<}(\alpha_m)\rangle \\ i^{n,m}(t+1,\vec{x},\vec{\alpha},\mathcal{P}) & = & s(i^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),C_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),C_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P})) \\ C_{\#}^{n,m}(t+1,\vec{x},\vec{\alpha},\mathcal{P}) & = & S_{\#}(i^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),C_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),C_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P})) \\ C_{*}^{n,m}(t+1,\vec{x},\vec{\alpha},\mathcal{P}) & = & S_{*}(i^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),C_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}),C_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P})) \end{array}$$

Por el Lema 63 tenemos que $i^{n,m}$, $C_{\#}^{n,m}$ y $C_{*}^{n,m}$ son $(\Sigma\cup\Sigma_p)$-p.r.. Ademas notese que
$$\begin{array}{rcl} E_{\#j}^{n,m} & = & \lambda t\vec{x}\vec{\alpha}\mathcal{P}\left[(C_{\#}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}))_j\right] \\ E_{*j}^{n,m} & = & \lambda t\vec{x}\vec{\alpha}\mathcal{P}\left[*^{<}((C_{*}^{n,m}(t,\vec{x},\vec{\alpha},\mathcal{P}))_j)\right] \end{array}$$
por lo cual las funciones $E_{\#j}^{n,m}$, $E_{*j}^{n,m}$, $j=1,2,...$, son $(\Sigma\cup\Sigma_p)$-p.r. □ Para $n,m\in\omega$ definamos la funcion $\Phi_{\#}^{n,m}$ de la siguiente manera:
$$\begin{array}{rcl} D_{\Phi_{\#}^{n,m}} & = & \left\{(\vec{x},\vec{\alpha},\mathcal{P})\in\omega^n\times\Sigma^{*m}\times\mathrm{Pro}^{\Sigma}:(\vec{x},\vec{\alpha})\in D_{\Psi_{\mathcal{P}}^{n,m,\omega}}\right\} \\ \Phi_{\#}^{n,m}(\vec{x},\vec{\alpha},\mathcal{P}) & = & \Psi_{\mathcal{P}}^{n,m,\omega}(\vec{x},\vec{\alpha}),\text{ para cada }(\vec{x},\vec{\alpha},\mathcal{P})\in D_{\Phi_{\#}^{n,m}} \end{array}$$

Similarmente, definamos la funcion $\Phi_{*}^{n,m}$ de la siguiente manera:
$$\begin{array}{rcl} D_{\Phi_{*}^{n,m}} & = & \left\{(\vec{x},\vec{\alpha},\mathcal{P})\in\omega^n\times\Sigma \right. \\ \Phi_{*}^{n,m}(\vec{x},\vec{\alpha},\mathcal{P}) & = & \Psi_{\mathcal{P}}^{n,m,\Sigma^*}(\vec{x},\vec{\alpha}),\text{ para} \end{array}$$

Notese que
$$\begin{array}{rcl} \Phi_{\#}^{n,m} & = & \lambda\vec{x}\vec{\alpha}\mathcal{P}\left[\Psi_{\mathcal{P}}^{n,m,\omega}(\vec{x},\vec{\alpha})\right] \\ \Phi_{*}^{n,m} & = & \lambda\vec{x}\vec{\alpha}\mathcal{P}\left[\Psi_{\mathcal{P}}^{n,m,\Sigma^*}(\vec{x},\vec{\alpha})\right] \end{array}$$

Teorema 65 Las funciones $\Phi_{\#}^{n,m}$ y $\Phi_{*}^{n,m}$ son $(\Sigma\cup\Sigma_p)$-recursivas. Prueba: Veremos que $\Phi_{\#}^{n,m}$ es $(\Sigma\cup\Sigma_p)$-recursiva. Sea H el predicado $(\Sigma\cup\Sigma_p)$-mixto

$\lambda t\vec{x}\vec{\alpha}\mathcal{P}\left[i^{n,m}(t,x_1,...,x_n,\alpha_1,...,\alpha_m,\mathcal{P})=n(\mathcal{P})+1\right]$.

Note que $D_H=\omega^{n+1}\times\Sigma^{*m}\times\mathrm{Pro}^{\Sigma}$. Ya que the functiones $i^{n,m}$ y $\lambda\mathcal{P}\left[n(\mathcal{P})\right]$ son $(\Sigma\cup\Sigma_p)$-p.r., H lo es. Notar que $D_{M(H)}=D_{\Phi_{\#}^{n,m}}$. Ademas para $(\vec{x},\vec{\alpha},\mathcal{P})\in D_{M(H)}$, tenemos que $M(H)(\vec{x},\vec{\alpha},\mathcal{P})$ es la menor cantidad de pasos necesarios para que $\mathcal{P}$ termine partiendo del estado $((x_1,...,x_n,0,0,...),(\alpha_1,...,\alpha_m,\varepsilon,\varepsilon,...))$. Ya que H es $(\Sigma\cup\Sigma_p)$-p.r., tenemos que $M(H)$ es $(\Sigma\cup\Sigma_p)$-r.. Notese que para $(\vec{x},\vec{\alpha},\mathcal{P})\in D_{M(H)}=D_{\Phi_{\#}^{n,m}}$ tenemos que $\Phi_{\#}^{n,m}(\vec{x},\vec{\alpha},\mathcal{P})=E_{\#1}^{n,m}(M(H)(\vec{x},\vec{\alpha},\mathcal{P}),\vec{x},\vec{\alpha},\mathcal{P})$

lo cual con un poco mas de trabajo nos permite probar que $\Phi_{\#}^{n,m}=E_{\#1}^{n,m}\circ\left(M(H),p_1^{n,m+1},...,p_{n+m+1}^{n,m+1}\right)$

Ya que la funcion $E_{\#1}^{n,m}$ es $(\Sigma\cup\Sigma_p)$-r., lo es $\Phi_{\#}^{n,m}$. □

Corolario 66 Si $f:D_f\subseteq\omega^n\times\Sigma^{*m}\to O$ es Σ-computable, entonces f es Σ-recursiva. Prueba: Haremos el caso $O=\Sigma^*$. Sea $\mathcal{P}_0$ un programa que compute a f. Primero veremos que f es $(\Sigma\cup\Sigma_p)$ -recursiva. Note que

$f=\Phi_{*}^{n,m}\circ\left(p_1^{n,m},...,p_{n+m}^{n,m},C_{\mathcal{P}_0}^{n,m}\right)$

donde cabe destacar que $p_1^{n,m},...,p_{n+m}^{n,m}$ son las proyecciones respecto del alfabeto $\Sigma\cup\Sigma_p$, es decir que tienen dominio $\omega^n\times(\Sigma\cup\Sigma_p)^{*m}$. Ya que $\Phi_{*}^{n,m}$ es $(\Sigma\cup\Sigma_p)$-recursiva tenemos que f lo es. O sea que el Teorema 51 nos dice que f es Σ -recursiva. □ El teorema anterior junto con el Teorema 56 nos garantizan que los conceptos de funcion Σ-recursiva y de funcion Σ -computable coinciden, es decir que los dos modelos matematicos de computabilidad efectiva que hemos estudiado, el funcional y el imperativo, coinciden. Como veremos en el proximo capitulo, el modelo introducido por Turing tambien resulta equivalente en el sentido de que una funcion Σ -mixta es computable por una maquina de Turing si y solo si es Σ -recursiva. Otro modelo matematico de computabilidad efectiva es el llamado lamda calculus, introducido por Church, el cual tambien resulta equivalente a los estudiados por nosotros. El hecho de que tan distintos paradigmas computacionales hayan resultado equivalentes hace pensar que en realidad los mismos han tenido exito en capturar la totalidad de las funciones Σ -efectivamente computables. Esta aseveracion es conocida como la

Tesis de Church: Toda funcion Σ-efectivamente computable es Σ-recursiva.

Si bien no se ha podido dar una prueba estrictamente matematica de la Tesis de Church, es un sentimiento comun de los investigadores del area que la misma es verdadera.

Un corolario interesante que se puede obtener del teorema anterior es que toda funcion Σ-recursiva puede obtenerse combinando las reglas basicas en una forma muy particular.

Corolario 67 Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to O$ es Σ-recursiva, entonces existe un predicado Σ-p.r. $P : \mathbf{N} \times \omega^n \times \Sigma^{*m} \to \omega$ y una funcion Σ-p.r. $g : \mathbf{N} \to O$ tales que $f = g \circ M(P)$. Prueba: Supongamos que $O = \Sigma^*$. Sea $\mathcal{P}_0$ un programa el cual compute a f. Sea $<$ un orden total estricto sobre Σ. Note que podemos tomar

$$\begin{aligned} P &= \lambda t\vec{x}\vec{\alpha}[i^{n,m}((t)_1, \vec{x}, \vec{\alpha}, \mathcal{P}_0) = n(\mathcal{P}_0) + 1 \wedge (t)_2 = \#^{<}(E_{*1}^{n,m}((t)_1, \vec{x}, \vec{\alpha}, \mathcal{P}_0))] \\ g &= \lambda t\,[*^{<}((t)_2)]\,. \end{aligned}$$

(Justifique por que P es Σ-p.r..) $\Box$

Extension del lema de division por casos

Usando las funciones $i^{n,m}$, $E_{\#}^{n,m}$ y $E_{*}^{n,m}$ podemos extender el lema de division por casos para funciones Σ -recursivas en general.

Lema 68 Supongamos $f_i : D_{f_i} \subseteq \omega^n \times \Sigma^{*m} \to O$, $i = 1, ..., k$, son funciones Σ-recursivas tales que $D_{f_i} \cap D_{f_j} = \varnothing$ para $i \neq j$. Entonces la funcion $f_1 \cup ... \cup f_k$ es Σ-recursiva. Prueba: Probaremos el caso $k = 2$ y $O = \Sigma^*$. Sean $\mathcal{P}_1$ y $\mathcal{P}_2$ programas que computen las funciones f_1 y f_2, respectivamente. Sean

$$\begin{aligned} P_1 &= \lambda t\vec{x}\vec{\alpha}\,[i^{n,m}(t, \vec{x}, \vec{\alpha}, \mathcal{P}_1) = n(\mathcal{P}_1) + 1] \\ P_2 &= \lambda t\vec{x}\vec{\alpha}\,[i^{n,m}(t, \vec{x}, \vec{\alpha}, \mathcal{P}_2) = n(\mathcal{P}_2) + 1] \end{aligned}$$

Notese que $D_{P_1} = D_{P_2} = \omega \times \omega^n \times \Sigma^{*m}$ y que P_1 y P_2 son $(\Sigma \cup \Sigma_p)$-p.r.. Ya que son Σ-mixtos, el Teorema 51 nos dice que son Σ-p.r.. Tambien notese que $D_{M((P_1 \vee P_2))} = D_{f_1} \cup D_{f_2}$. Definamos

$$\begin{aligned} g_1 &= \lambda\vec{x}\vec{\alpha}\left[E_{*1}^{n,m}(M\left((P_1 \vee P_2)\right)(\vec{x}, \vec{\alpha}), \vec{x}, \vec{\alpha}, \mathcal{P}_1)^{P_i(M((P_1 \vee P_2))(\vec{x},\vec{\alpha}),\vec{x},\vec{\alpha})}\right] \\ g_2 &= \lambda\vec{x}\vec{\alpha}\left[E_{*1}^{n,m}(M\left((P_1 \vee P_2)\right)(\vec{x}, \vec{\alpha}), \vec{x}, \vec{\alpha}, \mathcal{P}_2)^{P_i(M((P_1 \vee P_2))(\vec{x},\vec{\alpha}),\vec{x},\vec{\alpha})}\right] \end{aligned}$$

Notese que g_1 y g_2 son Σ-recursivas y que $D_{g_1} = D_{g_2} = D_{f_1} \cup D_{f_2}$, Ademas notese que

$$g_1(\vec{x}, \vec{\alpha}) = \begin{cases} f_1(\vec{x}, \vec{\alpha}) & \text{si } (\vec{x}, \vec{\alpha}) \in D_{f_1} \\ \varepsilon & \text{caso contrario} \end{cases}$$

$$g_2(\vec{x}, \vec{\alpha}) = \begin{cases} f_2(\vec{x}, \vec{\alpha}) & \text{si } (\vec{x}, \vec{\alpha}) \in D_{f_2} \\ \varepsilon & \text{caso contrario} \end{cases}$$

O sea que $f_1 \cup f_2 = \lambda\alpha\beta\,[\alpha\beta] \circ (g_1, g_2)$ es Σ-recursiva. $\Box$

El halting problem

Cuando $\Sigma \supseteq \Sigma_p$, podemos definir

$Halt^{\Sigma} = \lambda\mathcal{P}\left[(\exists t \in \omega)\; i^{0,1}(t, \mathcal{P}, \mathcal{P}) = n(\mathcal{P}) + 1\right]$.

Notar que el dominio de $Halt^{\Sigma}$ es Pro^{Σ} y que para cada $\mathcal{P} \in \text{Pro}^{\Sigma}$ tenemos que (*) $Halt(\mathcal{P}) = 1$ sii $\mathcal{P}$ se detiene partiendo del estado $((0, 0, ...), (\mathcal{P}, \varepsilon, \varepsilon, ...))$.

Lema 69 Supongamos $\Sigma \supseteq \Sigma_p$. Entonces $Halt^{\Sigma}$ es no Σ-recursivo. Prueba: Supongamos $Halt^{\Sigma}$ es Σ-recursivo y por lo tanto Σ -computable. Por la proposicion de existencia de macros tenemos que hay un macro

$\left[\text{IF } Halt^{\Sigma}(\text{W1}) \text{ GOTO A1}\right]$

Sea $\mathcal{P}_0$ el siguiente programa de $\mathcal{S}^{\Sigma}$ L1 $\left[\text{IF } Halt^{\Sigma}(\text{P1}) \text{ GOTO L1}\right]$

Note que - $\mathcal{P}_0$ termina partiendo desde $((0, 0, ...), (\mathcal{P}_0, \varepsilon, \varepsilon, ...))$ sii $Halt^{\Sigma}(\mathcal{P}_0) = 0$, lo cual produce una contradiccion si tomamos en (*) $\mathcal{P} = \mathcal{P}_0$. $\Box$

Conjuntos Σ-recursivamente enumerables

Dada una funcion $F : D_F \subseteq \omega^n \times \Sigma^{*m} \to \omega^k \times \Sigma^{*l}$ e $i \in \{1, ..., k + l\}$, usaremos F_i para denotar la funcion $p_i^{k,l} \circ F$. Notese que el dominio de cada F_i es igual a D_F. Un conjunto

$S\subseteq\omega^n\times\Sigma^{*m}$ es llamado Σ-recursivamente enumerable (Σ-r.e.) si $S=\varnothing$ o $S=I_F$, para alguna $F:\omega\rightarrow\omega^n\times\Sigma^{*m}$ tal que cada F_i es Σ-recursiva. Como puede notarse el concepto de conjunto Σ-recursivamente enumerable es la modelizacion matematica del concepto de conjunto Σ-efectivamente enumerable, dentro del paradigma funcional o Godeliano. Es decir

Teorema 70 Sea $S\subseteq\omega^n\times\Sigma^{*m}$. Entonces S es Σ-efectivamente enumerable sii S es Σ-recursivamente enumerable Prueba: ($\Rightarrow$) Use la Tesis de Church.

($\Leftarrow$) Use el Teorema 42. $\Box$

El siguiente teorema es el analogo recursivo del Teorema 17.

Teorema 71 Dado $S\subseteq\omega^n\times\Sigma^{*m}$, son equivalentes (1) S es Σ-recursivamente enumerable (2) $S=I_F$, para alguna $F:D_F\subseteq\omega^k\times\Sigma^{*l}\rightarrow\omega^n\times\Sigma^{*m}$ tal que cada F_i es Σ-recursiva. (3) $S=D_f$, para alguna funcion Σ-recursiva f (4) $S=\varnothing$ o $S=I_F$, para alguna $F:\omega\rightarrow\omega^n\times\Sigma^{*m}$ tal que cada F_i es Σ-p.r. Prueba: (2)$\Rightarrow$(3). Para $i=1,...,n+m$, sea $\mathcal{P}_i$ un programa el cual computa a F_i y sea $<$ un orden total estricto sobre Σ. Sea $P:\mathbf{N}\times\omega^n\times\Sigma^{*m}\rightarrow\omega$ dado por $P(t,\vec{x},\vec{\alpha})=1$ sii se cumplen las siguientes condiciones

$$\begin{array}{rcl}
i^{k,l}(((t)_{k+l+1},(t)_1,...,(t)_k,*^{<}((t)_{k+1}),...,*^{<}((t)_{k+l})),\mathcal{P}_1) & = & n(\mathcal{P}_1)+1\\
 & & \vdots\\
i\,((t)_{k+l+1},(t)_1...(t)_k,*^{<}((t)_{k+1})...\,*^{<}\,((t)_{k+l})),\mathcal{P}_{n+m}) & = & n(\mathcal{P}_{n+m})+1\\
E_{\#1}^{k,l}((t)_{k+l+1},(t)_1,...,(t)_k,*^{<}((t)_{k+1}),...,*^{<}((t)_{k+l})),\mathcal{P}_1) & = & x_1\\
 & & \vdots\\
E_{\#1}^{k,l}((t)_{k+l+1},(t)_1,...,(t)_k,*^{<}((t)_{k+1}),...,*^{<}((t)_{k+l})),\mathcal{P}_n) & = & x_n\\
E_{*1}^{k,l}((t)_{k+l+1},(t)_1,...,(t)_k,*^{<}((t)_{k+1}),...,*^{<}((t)_{k+l})),\mathcal{P}_{n+1}) & = & \alpha_1\\
 & & \vdots\\
E_{*1}^{k,l}((t)_{k+l+1},(t)_1,...,(t)_k,*^{<}((t)_{k+1}),...,*^{<}((t)_{k+l})),\mathcal{P}_{n+m}) & = & \alpha_m
\end{array}$$

Note que P es $(\Sigma\cup\Sigma_p)$-p.r. y por lo tanto P es Σ-p.r.. Pero entonces $M(P)$ es Σ-r. lo cual nos dice que se cumple (3) ya que $D_{M(P)}=I_F=S$. (3)$\Rightarrow$(4). Supongamos $S\neq\varnothing$. Sea $(z_1,...,z_n,\gamma_1,...,\gamma_m)\in S$ fijo. Sea $\mathcal{P}$ un programa el cual compute a f y sea $<$ un orden total estricto sobre Σ. Sea $P:\mathbf{N}\rightarrow\omega$ dado por $P(x)=1$ sii

$i^{n,m}\left((x)_{n+m+1},(x)_1,...,(x)_n,*^{<}((x)_{n+1}),...,*^{<}((x)_{n+m})),\mathcal{P}\right)=n(\mathcal{P})+1$

Es facil ver que P es $(\Sigma\cup\Sigma_p)$-p.r. por lo cual es Σ-p.r.. Sea $\bar{P}=P\cup C_0^{1,0}\mid_{\{0\}}$. Para $i=1,...,n$, definamos $F_i:\omega\rightarrow\omega$ de la siguiente manera $F_i(x)=\left\{\begin{array}{ccc}(x)_i & \text{si} & \bar{P}(x)=1\\ z_i & \text{si} & \bar{P}(x)\neq1\end{array}\right.$

Para $i=n+1,...,n+m$, definamos $F_i:\omega\rightarrow\Sigma^*$ de la siguiente manera $F_i(x)=\left\{\begin{array}{ll}*^{<}((x)_i) & \text{si}\quad\bar{P}(x)=1\\ \gamma_{i-n} & \text{si}\quad\bar{P}(x)\neq1\end{array}\right.$

Por el lema de division por casos, cada F_i es Σ-p.r.. Es facil ver que $F=(F_1,...,F_{n+m})$ cumple (4). $\Box$

Corolario 72 Supongamos $f:D_f\subseteq\omega^n\times\Sigma^{*m}\rightarrow O$ es Σ-recursiva y $S\subseteq D_f$ es Σ-r.e., entonces $f\mid_S$ es Σ-recursiva. Prueba: Supongamos $O=\Sigma^*$. Por el teorema anterior $S=D_g$, para alguna funcion Σ-recursiva g. Notese que componiendo adecuadamente podemos suponer que $I_g=\{\varepsilon\}$. O sea que tenemos $f\mid_S=\lambda\alpha\beta\,[\alpha\beta]\circ(f,g)$. $\Box$

Corolario 73 Supongamos $f:D_f\subseteq\omega^n\times\Sigma^{*m}\rightarrow O$ es Σ-recursiva y $S\subseteq I_f$ es Σ-r.e., entonces $f^{-1}(S)=\{(\vec{x},\vec{\alpha}):f(\vec{x},\vec{\alpha})\in S\}$ es Σ-r.e.. Prueba: Por el teorema anterior $S=D_g$, para alguna funcion Σ-recursiva g. O sea que $f^{-1}(S)=D_{g\circ f}$ es Σ-r.e.. $\Box$

Corolario 74 Supongamos $S_1,S_2\subseteq\omega^n\times\Sigma^{*m}$ son conjuntos Σ-r.e.. Entonces $S_1\cap S_2$ es Σ-r.e.. Prueba: Por el teorema anterior $S_i=D_{g_i}$, con g_1,g_2 funciones Σ-recursivas. Notese que podemos suponer que $I_{g_1},I_{g_2}\subseteq\omega$ por lo que $S_1\cap S_2=D_{\lambda xy[xy]\circ(g_1,g_2)}$ es Σ-r.e.. $\Box$

Corolario 75 Supongamos $S_1,S_2\subseteq\omega^n\times\Sigma^{*m}$ son conjuntos Σ-r.e.. Entonces $S_1\cup S_2$ es Σ-r.e. Prueba: Supongamos $S_1\neq\varnothing\neq S_2$. Sean $F,G:\omega\rightarrow\omega^n\times\Sigma^{*m}$ tales que $I_F=S_1$, $I_G=S_2$ y las funciones F_i's y G_i's son Σ-recursivas. Sean $f=\lambda x\,[Q(x,2)]$ y $g=\lambda x\,[Q(x\dot{-}1,2)]$. Sea

$H:\omega\rightarrow\omega^{n}\times\Sigma^{*m}$ dada por

$H_i = (F_i \circ f)|_{\{x:x \text{ es par}\}} \cup (G_i \circ g)|_{\{x:x \text{ es impar}\}}$

Por el Corolario 72 y el Lema 68, cada H_i es Σ-recursiva. Ya que $I_H = S_1 \cup S_2$.tenemos que $S_1 \cup S_2$ es Σ-r.e. $\Box$

Conjunto Σ-recursivo

Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ-recursivo si la funcion caracteristica de S,

$\chi_S : \omega^n \times \Sigma^{*m} \to \{0,1\}$

es Σ-recursiva. Como puede notarse el concepto de conjunto Σ -recursivo es la modelizacion matematica del concepto de conjunto Σ -efectivamente computable, dentro del paradigma funcional o Godeliano. Es decir

Teorema 76 Sea $S \subseteq \omega^n \times \Sigma^{*m}$. Entonces S es Σ-efectivamente computable sii S es Σ-recursivo Prueba: ($\Rightarrow$) Use la Tesis de Church.

($\Leftarrow$) Use el Teorema 42. $\Box$

Teorema 77 Sea $S \subseteq \omega^n \times \Sigma^{*m}$. Son equivalentes (a) S es Σ-recursivo (b) S y $(\omega^n \times \Sigma^{*m}) - S$ son Σ -recursivamente enumerables Prueba: (a)$\Rightarrow$(b). Note que $S = D_{Pred \circ \chi_S}$.

(b)$\Rightarrow$(a). Note que $\chi_S = C_1^{n,m}|_S \cup C_0^{n,m}|_{\omega^n \times \Sigma^{*m} - S}$. $\Box$

Recordemos que para el caso en que $\Sigma \supseteq \Sigma_p$, definimos

$Halt^\Sigma = \lambda\mathcal{P}\left[(\exists t \in \omega)\ i^{0,1}(t,\mathcal{P},\mathcal{P}) = n(\mathcal{P}) + 1\right]$

Lema 78 Supongamos que $\Sigma \supseteq \Sigma_p$. Entonces $A = \left\{\mathcal{P} \in \mathrm{Pro}^\Sigma : Halt^\Sigma(\mathcal{P})\right\}$

es Σ-r.e. y no es Σ-recursivo. Mas aun el conjunto $N = \left\{\mathcal{P} \in \mathrm{Pro}^\Sigma : \neg Halt^\Sigma(\mathcal{P})\right\}$ no es Σ-r.e. Prueba: Sea $P = \lambda t\mathcal{P}\left[i^{0,1}(t,\mathcal{P},\mathcal{P}) = n(\mathcal{P}) + 1\right]$. Note que P es Σ-p.r. por lo que $M(P)$ es Σ-r.. Ademas note que $D_{M(P)} = A$, lo cual implica que A es Σ-r.e.. Ya que $Halt^\Sigma$ es no Σ-recursivo (Lema 69) y

$Halt^\Sigma = C_1^{0,1} \mid_A \cup C_0^{0,1} \mid_N$

el Lema 68 nos dice que N no es Σ-r.e.. Finalmente supongamos A es Σ-recursivo. Entonces el conjunto $N = (\Sigma^* - A) \cap \mathrm{Pro}^\Sigma$

deberia serlo, lo cual es absurdo. $\Box$

0.5. Maquinas de Turing

En esta seccion desarrollaremos el paradigma de Turing de computabilidad efectiva. Primero veremos que el funcionamiento de las maquinas de Turing, tal como el de los programas de $\mathcal{S}^\Sigma$, puede ser completamente descripto via funciones primitivas recursivas respecto de un alfabeto suficientemente grande. Como corolario de esto obtendremos que las funciones Σ-mixtas que son computables via maquinas de Turing son Σ-recursivas. Luego veremos que todo programa puede ser simulado en forma natural por una maquina de Turing lo cual nos dara como corolario que toda funcion Σ-computable es computable por una maquina de Turing.

Una maquina de Turing es una 7-upla $M = (Q,\Sigma,\Gamma,\delta,q_0,B,F)$ donde

- Q es un conjunto finito cuyos elementos son llamados estados - Γ es un alfabeto que contiene a Σ - Σ es un alfabeto llamado el alfabeto de entrada - $B \in \Gamma - \Sigma$ es un simbolo de Γ llamado el blank symbol - $\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L,R,K\})$ - q_0 es un estado llamado el estado inicial de M - $F \subseteq Q$ es un conjunto de estados llamados finales Asumiremos siempre que Q es un alfabeto disjunto con Γ. Esto nos permitira dar definiciones matematicas precisas que formalizaran el funcionamiento de las maquinas de Turing en el contexto de las funciones mixtas.

Una descripcion instantanea sera una palabra de la forma $\alpha q\beta$, donde $\alpha,\beta \in \Gamma^*$, $[\beta]_{|\beta|} \neq B$ y $q \in Q$. La descripcion instantanea $\alpha_1...\alpha_n q\beta_1...\beta_m$, con $\alpha_1,...,\alpha_n,\ \beta_1,...,\beta_m \in \Gamma$, $n,m \geq 0$ representara la siguiente situacion

$$\begin{array}{cccccccccccc} \alpha_1 & \alpha_2 & ... & \alpha_n & \beta_1 & \beta_2 & ... & \beta_m & B & B & B & ... \\ & & & & \uparrow & & & & & & & \\ & & & & q & & & & & & & \end{array}$$

Usaremos Des para denotar el conjunto de las descripciones instantaneas. Definamos la funcion $St : Des \to Q$, de la siguiente manera $St(d) =$ unico simbolo de Q que ocurre en d

Dado $\alpha \in (Q \cup \Gamma)^*$, definamos $\lfloor \alpha \rfloor$ de la siguiente manera $\begin{array}{rcl} \lfloor \varepsilon \rfloor & = & \varepsilon \\ \lfloor \alpha\sigma \rfloor & = & \alpha\sigma, \text{ si } \sigma \neq B \\ \lfloor \alpha B \rfloor & = & \lfloor \alpha \rfloor \end{array}$

Es decir $\lfloor \alpha \rfloor$ es el resultado de remover de α el tramo final mas grande de la forma B^n. Recordemos que dada cualquier palabra α definimos

$$^\frown\alpha = \begin{cases} [\alpha]_2 ... [\alpha]_{|\alpha|} & \text{si} \quad |\alpha| \geq 2 \\ \varepsilon & \text{si} \quad |\alpha| \leq 1 \end{cases}$$

En forma similar definamos $\alpha^\frown = \begin{cases} [\alpha]_1 ... [\alpha]_{|\alpha|-1} & \text{si} \quad |\alpha| \geq 2 \\ \varepsilon & \text{si} \quad |\alpha| \leq 1 \end{cases}$

Dadas $d_1, d_2 \in Des$, escribiremos $d_1 \vdash d_2$ cuando existan $\sigma \in \Gamma$, $\alpha, \beta \in \Gamma^*$ y $p, q \in Q$ tales que se cumple alguno de los siguientes casos Caso 1.

$$\begin{array}{rcl} d_1 & = & \alpha p \beta \\ (q, \sigma, R) & \in & \delta\left(p, [\beta B]_1\right) \\ d_2 & = & \alpha\sigma q ^\frown\beta \end{array}$$

Caso 2.

$$\begin{array}{rcl} d_1 & = & \alpha p \beta \\ (q, \sigma, L) & \in & \delta\left(p, [\beta B]_1\right) \text{ y } \alpha \neq \varepsilon \\ d_2 & = & \left\lfloor \alpha^\frown q \, [\alpha]_{|\alpha|} \, \sigma ^\frown\beta \right\rfloor \end{array}$$

Caso 3.

$$\begin{array}{rcl} d_1 & = & \alpha p \beta \\ (q, \sigma, K) & \in & \delta(p, [\beta B]_1) \\ d_2 & = & \lfloor \alpha q \sigma ^\frown\beta \rfloor \end{array}$$

Escribiremos $d \not\vdash d'$ para expresar que no se da $d \vdash d'$. Para $d, d' \in Des$ y $n \geq 0$, escribiremos $d \overset{n}{\vdash} d'$ si existen $d_1, ..., d_{n+1} \in Des$ tales que $\begin{array}{rcl} d & = & d_1 \\ d' & = & d_{n+1} \\ d_i & \vdash & d_{i+1}, \text{ para } i = 1, ..., n. \end{array}$

Notese que $d \overset{0}{\vdash} d'$ sii $d = d'$. Finalmente definamos $d \overset{*}{\vdash} d'$ sii $(\exists n \in \omega)\ d \overset{n}{\vdash} d'$.

Diremos que una palabra $w \in \Sigma^*$ es aceptada por M cuando $\lfloor q_0 B w \rfloor \overset{*}{\vdash} d$, con d tal que $St(d) \in F$.

El lenguage aceptado por M se define de la siguiente manera $L(M) = \{w \in \Sigma^* : w \text{ es aceptada por } M\}$.

Dada $d \in Des$, diremos que M se detiene partiendo de d si existe $d' \in Des$ tal que - $d \overset{*}{\vdash} d'$ - $d' \not\vdash d''$, para cada $d'' \in Des$ Deberia quedar claro que es posible que $\alpha p \beta \not\vdash d$, para cada descripcion instantanea d, y que $\delta(p, [\beta B]_1)$ sea no vacio. Definamos

$H(M) = \{w \in \Sigma^* : M \text{ se detiene partiendo de } \lfloor q_0 B w \rfloor\}$

Lema 79 Sea $L \subseteq \Sigma^*$. entonces $L = L(M)$ para alguna maquina de Turing M sii $L = H(M)$ para alguna maquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$. Prueba: $(\Rightarrow)$ Dada una maquina $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, costruiremos una maquina $M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, \tilde{q}_0, B, \varnothing)$ tal que $L(M) = H(M_1)$. Tomaremos $\Gamma_1 = \Gamma \cup \{X\}$, con X un simbolo nuevo no perteneciente a Γ. Para cada $a \in \Sigma$, sea q_a un estado nuevo, no perteneciente a Q. Sean $\tilde{q}_0, q_r, q_d, q_B$ estados nuevos no pertenecientes a Q. Tomemos entonces

$Q_1 = Q \cup \{\tilde{q}_0, q_r, q_d, q_B\} \cup \{q_a : a \in \Sigma\}$

Finalmente definamos δ_1 de la siguiente manera:
$$\begin{array}{rcl}
\delta_1(\tilde{q}_0, B) & = & \{(q_B, X, R)\} \\
\delta_1(q_B, a) & = & \{(q_a, B, R)\}, \text{ para } a \in \Sigma \\
\delta_1(q_B, B) & = & \{(q_0, B, K)\} \\
\delta_1(q_a, b) & = & \{(q_b, a, R)\}, \text{ para } a, b \in \Sigma \\
\delta_1(q_a, B) & = & \{(q_r, a, L)\}, \text{ para } a \in \Sigma \\
\delta_1(q_r, a) & = & \{(q_r, a, L)\}, \text{ para } a \in \Sigma \\
\delta_1(q_r, B) & = & \{(q_0, B, K)\} \\
\delta_1(q, X) & = & \{(q, X, K)\}, \text{ para } q \in Q \\
\delta_1(q, \sigma) & = & \delta(q, \sigma) \cup \{(q_d, \sigma, K)\} \text{ , para } q \in F \text{ y } \sigma \\
\delta_1(q, \sigma) & = & \delta(q, \sigma), \text{ para } q \in Q - F \text{ y } \sigma \in \Gamma \\
\delta_1(q_d, \sigma) & = & \varnothing, \text{ para } \sigma \in \Gamma
\end{array}$$

(δ_1 se define igual a vacio para los casos no contemplados arriba). ($\Leftarrow$) Dada $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, dejamos al lector la construccion de una maquina $M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, \tilde{q}_0, B, \varnothing)$ tal que $H(M) = L(M_1)$. $\Box$

Lema 80 El predicado $\lambda ndd' \left[d \vdash d'\right]$ es $(\Gamma \cup Q)$-p.r.. Prueba: Note que $D_{\lambda dd'[d \vdash d']} = Des \times Des$. Tambien notese que los predicados

$\lambda p\sigma q\gamma \left[(p, \sigma, L) \in \delta(q, \gamma)\right]$

$\lambda p\sigma q\gamma \left[(p, \sigma, R) \in \delta(q, \gamma)\right]$

$\lambda p\sigma q\gamma \left[(p, \sigma, K) \in \delta(q, \gamma)\right]$

son $(\Gamma \cup Q)$-p.r. ya que los tres tienen dominio igual a $Q \times \Gamma \times Q \times \Gamma$ el cual es finito (Corolario 36). Sea $P_R : Des \times Des \times \Gamma \times \Gamma^* \times \Gamma^* \times Q \times Q \to \omega$ definido por $P_R(d, d', \sigma, \alpha, \beta, p, q) = 1$ sii $d = \alpha p \beta \wedge (q, \sigma, R) \in \delta\left(p, [\beta B]_1\right) \wedge d' = \alpha\sigma q \frown \beta$

Sea $P_L : Des \times Des \times \Gamma \times \Gamma^* \times \Gamma^* \times Q \times Q \to \omega$ definido por $P_L(d, d', \sigma, \alpha, \beta, p, q) = 1$ sii $d = \alpha p \beta \wedge (q, \sigma, L) \in \delta\left(p, [\beta B]_1\right) \wedge \alpha \neq \varepsilon \wedge d' = \left\lfloor \alpha \frown q \left[\alpha\right]_{|\alpha|} \sigma \frown \beta \right\rfloor$

Sea $P_K : Des \times Des \times \Gamma \times \Gamma^* \times \Gamma^* \times Q \times Q \to \omega$ definido por $P_K(d, d', \sigma, \alpha, \beta, p, q) = 1$ sii $d = \alpha p \beta \wedge (q, \sigma, K) \in \delta\left(p, [\beta B]_1\right) \wedge d' = \left\lfloor \alpha q \sigma \frown \beta \right\rfloor$

Se deja al lector la verificacion de que estos predicados son $(\Gamma \cup Q)$-p.r.. Notese que $\lambda dd' \left[d \vdash d'\right]$ es igual al predicado $\lambda dd' \left[(\exists \sigma \in \Gamma)(\exists \alpha, \beta \in \Gamma^*)(\exists p, q \in Q)(P_R \vee P_L \vee P_K)(d, d', \sigma, \alpha, \beta, p, q)\right]$

lo cual por el Lema 39 nos dice que $\lambda dd' \left[d \vdash d'\right]$ es $(\Gamma \cup Q)$-p.r. $\Box$

Proposición 81 $\lambda ndd' \left[d \overset{n}{\vdash} d'\right]$ es $(\Gamma \cup Q)$-p.r.. Prueba: Sea $Q = \lambda dd' \left[d \vdash d'\right] \cup C_0^{0,2} \mid_{(\Gamma \cup Q)^{*2} - Des^2}$ es decir Q es el resultado de extender con el valor 0 al predicado $\lambda dd' \left[d \vdash d'\right]$ de manera que este definido en todo $(\Gamma \cup Q)^{*2}$. Sea $<$ un orden total estricto sobre $\Gamma \cup Q$ y sea $Q_1 : \mathbf{N} \times Des \times Des \to \omega$ definido por $Q_1(x, d, d') = 1$ sii

$$\left((\forall i \in \mathbf{N})_{i \leq Lt(x)} *^<((x)_i) \in Des\right) \wedge *^<((x)_1) = d \wedge$$
$$*^<((x)_{Lt(x)}) = d' \wedge \left((\forall i \in \mathbf{N})_{i \leq Lt(x) \dot{-} 1}\ Q(*^<((x)_i), *^<((x)_{i+1}))\right)$$

Notese que dicho rapidamente $Q_1(x, d, d') = 1$ sii x codifica una computacion que parte de d y llega a d'. Se deja al lector la verificacion de que este predicado es $(\Gamma \cup Q)$-p.r.. Notese que

$$\lambda ndd' \left[d \overset{n}{\vdash} d'\right] = \lambda ndd' \left[(\exists x \in \mathbf{N})\ Lt(x) = n + 1 \wedge Q_1(x, d, d')\right]$$

Es decir que solo nos falta acotar el cuantificador existencial, para poder aplicar el lema de cuantificacion acotada. Ya que cuando $d_1, ..., d_{n+1} \in Des$ son tales que $d_1 \vdash d_2 \vdash ... \vdash d_{n+1}$ tenemos que $|d_i| \leq |d_1| + n$, para $i = 1, ..., n$

una posible cota para dicho cuantificador es $\displaystyle\prod_{i=1}^{n+1} pr(i)^{|\Gamma \cup Q|^{|d|+n}}$.

O sea que, por el lema de cuantificacion acotada, tenemos que el predicado $\lambda ndd' \left[d \overset{n}{\vdash} d'\right]$ es $(\Gamma \cup Q)$-p.r. $\Box$

Teorema 82 Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ una maquina de Turing. Entonces $L(M)$ es Σ-recursivamente enumerable. Prueba: Sea P el siguiente predicado $(\Gamma \cup Q)$-mixto

$\lambda n\alpha\left[(\exists d \in Des)\ \lfloor q_0 B\alpha \rfloor \overset{n}{\vdash} d \wedge St(d) \in F\right]$

Notese que $D_P = \omega \times \Gamma^*$. Dejamos al lector probar que P es $(\Gamma \cup Q)$-p.r.. Sea $P' = P\mid_{\omega \times \Sigma^*}$. Notese que $P'(n, \alpha) = 1$ sii $\alpha \in L(M)$ atestiguado por una computacion de longitud n. Ya que P' es $(\Gamma \cup Q)$-p.r. (por que?) y ademas es Σ-mixto, el Teorema 51 nos dice que P' es Σ-p.r.. Ya que $L(M) = D_{M(P')}$, el Teorema 71 nos dice que $L(M)$ es Σ-r.e.. $\square$

0.5.1. Funciones Σ-Turing computables

Para poder computar funciones mixtas con una maquina de Turing necesitaremos un simbolo para representar numeros sobre la cinta. Llamaremos a este simbolo unit y lo denotaremos con $\shortmid$. Mas formalmente una maquina de Turing con unit es una 8-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ tal que $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$ es una maquina de Turing y $\shortmid$ es un simbolo distingido perteneciente a $\Gamma - (\{B\} \cup \Sigma)$.

Una maquina de Turing M sera llamada deterministica cuando se de que $|\delta(p, \sigma)| \leq 1$, cualesquiera sean $p \in Q$ y $\sigma \in \Gamma$.

Diremos que una funcion $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ es Σ-Turing computable si existe una maquina de Turing deterministica con unit, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ tal que:

(1) Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces hay un $p \in Q$ tal que $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B\alpha_1 B...B\alpha_m \rfloor \overset{*}{\vdash} \lfloor pBf(\vec{x}, \vec{\alpha}) \rfloor$

y $\lfloor pBf(\vec{x}, \vec{\alpha}) \rfloor \nvdash d$, para cada $d \in Des$ (2) Si $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} - D_f$, entonces M no se detiene partiendo de $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B\alpha_1 B...B\alpha_m \rfloor$.

En forma similar, una funcion $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$, es llamada Σ- Turing computable si existe una maquina de Turing deterministica con unit, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$, tal que:

(1) Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces hay un $p \in Q$ tal que $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B\alpha_1 B...B\alpha_m \rfloor \overset{*}{\vdash} \left\lfloor pB \shortmid^{f(\vec{x}, \vec{\alpha})} \right\rfloor$

y $\left\lfloor pB \shortmid^{f(\vec{x}, \vec{\alpha})} \right\rfloor \nvdash d$, para cada $d \in Des$ (2) Si $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} - D_f$, entonces M no se detiene partiendo de $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_n} B\alpha_1 B...B\alpha_m \rfloor$

Cuando M y f cumplan los items (1) y (2) de la definicion anterior, diremos que la funcion f es computada por M.

Teorema 83 Supongamos $f : S \subseteq \omega^n \times \Sigma^{*m} \to O$ es Σ-Turing computable. Entonces f es Σ-recursiva. Prueba: Supongamos $O = \Sigma^*$ y sea $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ una maquina de Turing deterministica con unit la cual compute a f. Sea $<$ un orden total estricto sobre $\Gamma \cup Q$. Sea $P : \mathbf{N} \times \omega^n \times \Sigma^{*m} \to \omega$ dado por $P(x, \vec{x}, \vec{\alpha}) = 1$ sii

$(\exists q \in Q)\ \lfloor q_0 B \shortmid^{x_1} ...B \shortmid^{x_n} B\alpha_1 ...B\alpha_m \rfloor \overset{(x)_1}{\vdash} \lfloor qB *^{<} ((x)_2) \rfloor \wedge$ $\wedge(\forall$

$Des)_{|d| \leq |*^{<}((x)_2)|+2}\ \lfloor qB *^{<} ((x)_2) \rfloor \nvdash d$ Es facil ver que P es $(\Gamma \cup Q)$-p.r. por lo que P es Σ-p.r. ya que es Σ-mixto. Notese que

$f = \lambda \vec{x} \vec{\alpha} \left[\left(\min_x P(x, \vec{x}, \vec{\alpha}) \right)_2 \right]$,

lo cual nos dice que f es Σ-recursiva. $\square$ A continuacion nos proponemos probar que el paradigma de las maquinas de Turing es completo. El siguiente lema es clave ya que nos muestra como pueden simularse en forma muy natural, los programas de $\mathcal{S}^\Sigma$ con maquinas de Turing deterministicas

Lema 84 Sea $\mathcal{P} \in \mathrm{Pro}^\Sigma$ y sea k tal que las variables que ocurren en $\mathcal{P}$ estan todas en la lista $\mathrm{N}1, ..., \mathrm{N}\bar{k}, \mathrm{P}1, ..., \mathrm{P}\bar{k}$. Para cada $a \in \Sigma \cup \{\shortmid\}$, sea $\tilde{a}$ un nuevo simbolo. Sea $\Gamma = \Sigma \cup \{B, \shortmid\} \cup \{\tilde{a} : a \in \Sigma \cup \{\shortmid\}\}$. Entonces hay una maquina de Turing deterministica con unit $M = (Q, \Gamma, \Sigma, \delta, q_0, B, \shortmid, \{q_f\})$ la cual satisface (1) $\delta(q_f, \sigma) = \varnothing$, 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 $((x_1, ..., x_k, 0, ...), (\alpha_1, ..., \alpha_k, \varepsilon, ...))$

sii M se detiene partiendo de la descripcion instantanea $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_k} B\alpha_1 B...B\alpha_k B \rfloor$

(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 $((x_1, ..., x_k, 0, ...), (\alpha_1, ...,$

y llega al estado $((y_1, ..., y_k, 0, ...), (\beta_1, ..., \beta_k, \varepsilon, ...))$

entonces $\lfloor q_0 B \shortmid^{x_1} B...B \shortmid^{x_k} B\alpha_1 B...B\alpha_k B \rfloor \overset{*}{\vdash} \lfloor q_f B \shortmid^{y_1} B...B \shortmid^{y_k} B\beta_1 B...B\beta_k B \rfloor$ Prueba: Dado un estado $((x_1, ..., x_k, 0, ...), (\alpha_1, ..., \alpha_k, \varepsilon, ...))$ lo representaremos en la cinta de la siguiente manera

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

A continuacion describiremos una serie de maquinas las cuales simularan, via la representacion anterior, las distintas clases de instrucciones que pueden ocurrir en $\mathcal{P}$. Todas las maquinas definidas tendran a $|$ como unit y a B como blanco, tendran a Σ como su alfabeto terminal y su alfabeto mayor sera $\Gamma = \Sigma \cup \{B, |\} \cup \{\tilde{a} : a \in \Sigma \cup \{|\}\}$. Ademas tendran uno o dos estados finales con la propiedad de que si q es un estado final, entonces $\delta(q,\sigma) = \varnothing$, 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 maquina tal que

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

Es claro que la maquina M_i^+ simula la instruccion $\mathrm{N}\bar{\imath} \leftarrow \mathrm{N}\bar{\imath} + 1$. Para $1 \leq i \leq k$, sea $M_i^{\dot{-}}$ una maquina tal que

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

Para $1 \leq i \leq k$ y $a \in \Sigma$, sea M_i^a una maquina tal que $\begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... \\ \uparrow \\ q_f \end{array}$

Para $1 \leq i \leq k$, sea $M_i^{\frown}$ una maquina tal que $\begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_{i-1} B ... \\ \uparrow \\ q_f \end{array}$

Para $j = 1, ..., k$, y $a \in \Sigma$, sea IF_j^a una maquina con dos estados finales q_{si} y q_{no} tal que 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 $\begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_k \\ \uparrow \\ q_{no} \end{array}$

Analogamente para $j = 1, ..., k$, sea IF_j una maquina tal que si $x_j \neq 0$, entonces $\begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... \\ \uparrow \\ q_0 \end{array}$

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 $1 \leq i, j \leq k$, sea $M_{i \leftarrow j}^*$ una maquina tal que $\begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_{i-} \\ \uparrow \\ q_f \end{array}$

Para $1 \leq i, j \leq k$, sea $M_{i \leftarrow j}^{\#}$ una maquina tal que $\begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B\,|^{x_1} ... B\,|^{x_{i-1}}\, B\,|^{x_j}\, B\,|^{x_i} \\ \uparrow \\ q_f \end{array}$

Para $1 \leq i \leq k$, sea $M_{i \leftarrow 0}$ una maquina tal que $\begin{array}{l} B\,|^{x_1} ... B\,|^{x_k}\, B\alpha_1 ... B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B\,|^{x_1} ... B\,|^{x_{i-1}}\, BB\,|^{x_{i+1}} ... \\ \uparrow \\ q_f \end{array}$

Para $1 \leq i \leq k$, sea $M_{i\leftarrow\varepsilon}$ una maquina tal que $\begin{array}{l} B\mid^{x_1}...B\mid^{x_k}B\alpha_1...B\alpha_k \\ \uparrow \\ q_0 \end{array} \overset{*}{\vdash} \begin{array}{l} B\mid^{x_1}...B\mid^{x_k}B\alpha_1...B\alpha_{i-1}B \\ \uparrow \\ q_f \end{array}$

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 $\delta = \varnothing$ en cualquier otro caso. Finalmente sea
$M_{\text{GOTO}} = (\{q_0, q_{si}, q_{no}\}, \Gamma, \Sigma, \delta, q_0, B, \mid, \{q_{si}, q_{no}\})$,
con $\delta(q_0, B) = \{(q_{si}, B, K)\}$ y $\delta = \varnothing$ en cualquier otro caso. Para poder hacer concretamente las maquinas recien descriptas deberemos dise ñar antes algunas maquinas auxiliares. Para cada $j \geq 1$, sea D_j la maquina descripta en la Figura 1. Notese que

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

siempre que $\alpha, \gamma \in \Gamma^*$, $\beta_1, ..., \beta_j \in (\Gamma - \{B\})^*$. Analogamente tenemos definidas las maquinas I_j. Para $j \geq 1$, sea TD_j una maquina con un solo estado final q_f y tal que

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

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 bloque γ y agrega un blanco en el espacio que se genera a la izquierda de dicho bloque. Por ejemplo, para el caso de $\Sigma = \{\&\}$ podemos tomar TD_3 igual a la maquina de la Figura 3. Analogamente, para $j \geq 1$, sea TI_j una maquina tal que

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

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 bloque γ (por lo cual en el lugar de σ queda el primer simbolo de γ). Teniendo las maquinas auxiliares antes definidas podemos combinarlas para obtener las maquinas simuladoras de instrucciones. Por ejemplo M_i^a puede ser la maquina descripta en la Figura 4. En la Figura 2 tenemos una posible forma de diseñar la maquina IF_i^a. En la Figura 7 tenemos una posible forma de diseñar la maquina $M_{i\leftarrow j}^*$ para el caso $\Sigma = \{a, b\}$ y $i < j$.

Supongamos ahora que $\mathcal{P} = I_1...I_n$. Para cada $i = 1, ..., n$, definiremos una maquina M_i que simulara la instruccion I_i. Luego uniremos adecuadamente estas maquinas para formar la maquina que simulara a $\mathcal{P}$

- Si $Bas(I_i) = \text{N}\bar{j} \leftarrow \text{N}\bar{j} + 1$ tomaremos $M_i = M_j^+$ - Si $Bas(I_i) = \text{N}\bar{j} \leftarrow \text{N}\bar{j} \dot{-} 1$ tomaremos $M_i = M_j^{\dot{-}}$ - Si $Bas(I_i) = \text{N}\bar{j} \leftarrow 0$ tomaremos $M_i = M_{j\leftarrow 0}$. - Si $Bas(I_i) = \text{N}\bar{j} \leftarrow \text{N}\bar{m}$ tomaremos $M_i = M_{j\leftarrow m}^{\#}$. - Si $Bas(I_i) = \text{IF N}\bar{j} \neq 0 \text{ GOTO L}\bar{m}$ tomaremos $M_i = IF_j$. - Si $Bas(I_i) = \text{P}\bar{j} \leftarrow \text{P}\bar{j}.a$ tomaremos $M_i = M_j^a$. - Si $Bas(I_i) = \text{P}\bar{j} \leftarrow {}^\frown\text{P}\bar{j}$ tomaremos $M_i = M_j^\frown$. - Si $Bas(I_i) = \text{P}\bar{j} \leftarrow \varepsilon$ tomaremos $M_i = M_{j\leftarrow\varepsilon}$. - Si $Bas(I_i) = \text{P}\bar{j} \leftarrow \text{P}\bar{m}$ tomaremos $M_i = M_{j\leftarrow m}^*$. - Si $Bas(I_i) = \text{IF P}\bar{j} \text{ BEGINS } a \text{ GOTO L}\bar{m}$ tomaremos $M_i = IF_j^a$. - Si $Bas(I_i) = \text{SKIP}$ tomaremos $M_i = M_{\text{SKIP}}$. - Si $Bas(I_i) = \text{GOTO L}\bar{m}$ tomaremos $M_i = M_{\text{GOTO}}$. Ya que la maquina M_i puede tener uno o dos estados finales, la representaremos como se muestra en la 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 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 $M_1, ..., M_n$ como lo muestra la Figura 6. Luego para cada i tal que $Bas(I_i)$ es de la forma $\alpha\text{GOTO L}\bar{m}$, 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 L$\bar{m}$. Es intuitivamente claro que la maquina asi obtenida cumple con lo requerido aunque una prueba formal de esto puede resultar extremadamente tediosa. $\Box$ Usando el lema anterior podemos probar que el paradigma computacional de Turing es completo.

Teorema 85 Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ es Σ-recursiva, entonces f es Σ-Turing computable. Prueba: Supongamos $O = \Sigma^*$. Ya que f es Σ-computable, existe $\mathcal{P} \in \text{Pro}^\Sigma$ el cual computa f. Note que podemos suponer que $\mathcal{P}$ tiene la propiedad de que cuando $\mathcal{P}$ termina, en el estado alcansado las variables numericas tienen todas el valor 0 y las alfabeticas distintas de P1 todas el valor ε. Sea M la maquina de Turing con unit dada por el lema anterior, donde elejimos el numero k con la propiedad adicional de ser mayor que n y m. Sea M_1 una maquina tal que para cada $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m}$,

$\lfloor q_0 B \mid^{x_1} B...B \mid^{x_n} B\alpha_1 B...B\alpha_n B \rfloor \overset{*}{\vdash} \left\lfloor qB \mid^{x_1} B...B \mid^{x_n} B^{k-n} B\alpha_1 B...B\alpha_m B \right\rfloor$

donde q_0 es el estado inicial de M_1 y q es un estado tal que $\delta(q, \sigma) = \varnothing$, para cada σ. Sea M_2 una maquina tal que para cada $\alpha \in \Sigma^*$, $\left\lfloor q_0 B^{k+1} \alpha \right\rfloor \overset{*}{\vdash} \lfloor qB\alpha \rfloor$

donde q_0 es el estado inicial de M_2 y q es un estado tal que $\delta(q, \sigma) = \varnothing$, para cada σ. Note que la concatenacion de M_1, M y M_2 (en ese orden) produce una maquina de Turing la cual computa f. $\Box$

Teorema 86 Si $L \subseteq \Sigma^*$ es Σ-r.e., entonces $L = L(M) = H(M)$ para alguna maquina de Turing deterministica M. Prueba: Por el Teorema 71 hay una funcion $f : L \rightarrow \omega$, la cual es Σ-recursiva. Sea $\mathcal{P}$ un programa el cual compute a f. Sea M la maquina de Turing deterministica dada en el lema anterior. Sea M_1 una maquina de Turing deterministica tal que para todo $\alpha \in \Sigma^*$,

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

donde q_0 es el estado inicial de M y q es un estado tal que $\delta(q, \sigma) = \varnothing$, para cada σ. Note que la concatenacion de M_1 con M (en ese orden) produce una maquina de Turing deterministica M_2 tal que $H(M_2) = L(M_2) = L$. $\Box$

Bibliografía

[1] Diego Vaggione, «Apunte de Clase, 2017», *FaMAF, UNC*.

[2] Agustín Curto, «Carpeta de Clase, 2017», *FaMAF, UNC*.