Lenguajes Formales y Computabilidad

Teoremas: Combo 1

Nicolás Cagliero

June 27, 2025

Proposición (Caracterización de conjuntos Σ-p.r.). *Un conjunto S es Σ-p.r. sii S es el dominio de alguna función Σ-p.r.*

(En la inducción de la prueba hacer solo el caso de la composición)

Proof. ($\Rightarrow$) Note que $S = D_{\mathrm{Pred}_S^{\omega^n \times \Sigma^{*m}}}$.

($\Leftarrow$) Probaremos por inducción en k que D_F es Σ-p.r., para cada $F \in \mathrm{PR}_k^\Sigma$.

El caso $k = 0$ es fácil, $F \in \{Suc,\ Pred,\ C_0^{0,0},\ C_\varepsilon^{0,0}\} \cup \{d_a : a \in \Sigma\} \cup \{p_j^{n,m} : 1 \leq j \leq n+m\}$ y los dominios de estas funciones son claramente Σ-p.r.. 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.

Supongamos que $F = g \circ [g_1, ..., g_r]$ con $g, g_1, ..., g_r \in \mathrm{PR}_k^\Sigma$. Si $F = \emptyset$, entonces es claro que $D_F = \emptyset$ es Σ-p.r. Supongamos entonces que F no es la función vacía. Tenemos entonces que r es de la forma $n + m$ y

$$g : D_g \subseteq \omega^n \times \Sigma^{*m} \to O$$

$$g_i : D_{g_i} \subseteq \omega^k \times \Sigma^{*l} \to \omega, \quad i = 1, ..., n$$

$$g_i : D_{g_i} \subseteq \omega^k \times \Sigma^{*l} \to \Sigma^*, \quad i = n+1, ..., n+m$$

con $O \in \{\omega, \Sigma^*\}$ y $k, l \in \omega$. Por Lema 19 (Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to O$ es Σ-pr. entonces existe una función Σ-pr, $\bar{f} : \omega^n \times \Sigma^{*m} \to O$ tal que $f = \bar{f}|_{D_f}$), hay funciones Σ-p.r. $\bar{g}_1, ..., \bar{g}_{n+m}$ las cuales son Σ-totales y cumplen

$$g_i = \bar{g}_i|_{D_{g_i}}, \quad \text{para } i = 1, ..., n+m.$$

Por hipótesis 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. Nótese que

$$\chi_{D_F}^{\omega^k\times\Sigma^l} = \left(\chi_{D_g}^{\omega^n\times\Sigma^{*m}} \circ [\bar{g}_1,...,\bar{g}_{n+m}] \wedge \chi_S^{\omega^k\times\Sigma^l}\right)$$

lo cual nos dice que D_F es Σ-p.r. ■

Teorema (Neumann vence a Gödel). *Si h es Σ-recursiva, entonces h es Σ-computable*

(En la inducción de la prueba hacer solo el caso $h = R(f,\mathcal{G})$, con $I_h \subseteq \omega$)

proof. Probaremos por inducción en k que

$$(*) \quad \text{Si } h \in \mathrm{R}_k^\Sigma, \text{ entonces } h \text{ es } \Sigma\text{-computable.}$$

El caso $k=0$: $h \in \{Suc,\ Pred,\ C_0^{0,0},\ C_\varepsilon^{0,0}\} \cup \{d_a : a \in \Sigma\} \cup \{p_j^{n,m} : 1 \leq j \leq n+m\}$. Los programas que computan cada función son:

$$Pred = \Psi_{\mathcal{P}}^{1,0,\#}$$

$$Suc = \Psi_{\mathrm{N1}\leftarrow\mathrm{N1}+1}^{1,0,\#}$$

$$C_0^{0,0} = \Psi_{\mathrm{SKIP}}^{0,0,\#}$$

$$C_\varepsilon^{0,0} = \Psi_{\mathrm{SKIP}}^{0,0,*}$$

$$d_a = \Psi_{\mathrm{P1}\leftarrow\mathrm{P1}.a}^{1,0,*}$$

$$p_i^{n,m} = \Psi_{\mathrm{N1}\leftarrow\mathrm{N}\bar{i}}^{n,m,\#} \text{ para } i \in \{1,\dots,n\}$$

$$p_i^{n,m} = \Psi_{\mathrm{P1}\leftarrow\mathrm{P}\overline{i-n}}^{n,m,*} \text{ para } i \in \{n+1,\dots,n+m\}$$

Supongamos que $(*)$ vale para k, veremos que vale para $k+1$. Sea $h \in \mathrm{R}_{k+1}^\Sigma - \mathrm{R}_k^\Sigma$. Hay varios casos...
Supongamos $h = R(f,\mathcal{G})$, con

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

$$g_a : \omega \times S_1 \times \ldots \times S_n \times L_1 \times \ldots \times L_m \times \Sigma^* \to \omega, \quad a \in \Sigma$$

elementos de R_k^Σ. Sea $\Sigma = \{a_1,...,a_r\}$. Por hipótesis inductiva, las funciones f, g_a, $a \in \Sigma$, son Σ-computables y por lo tanto tenemos macros

$$[\mathrm{V}\overline{n+1} \leftarrow f(\mathrm{V}1,...,\mathrm{V}\bar{n},\mathrm{W}1,...,\mathrm{W}\bar{m})]$$

$$[\mathrm{V}\overline{n+2} \leftarrow g_{a_i}(\mathrm{V}1,...,\mathrm{V}\bar{n},\mathrm{V}\overline{n+1},\mathrm{W}1,...,\mathrm{W}\bar{m},\mathrm{W}\overline{m+1})], \quad i=1,...,r$$

Podemos entonces hacer el siguiente programa:

$$
\begin{array}{ll}
 & [\mathrm{N}\overline{n+1} \leftarrow f(\mathrm{N}1,...,\mathrm{N}\overline{n},\mathrm{P}1,...,\mathrm{P}\overline{m})] \\
\mathrm{L}\overline{r+1} & \mathrm{IF\ P}\overline{m+1}\ \mathrm{BEGINS}\ a_1\ \mathrm{GOTO\ L1} \\
 & \vdots \\
 & \mathrm{IF\ P}\overline{m+1}\ \mathrm{BEGINS}\ a_r\ \mathrm{GOTO\ L}\overline{r} \\
 & \mathrm{GOTO\ L}\overline{r+2} \\
 & \\
\mathrm{L1} & \mathrm{P}\overline{m+1} \leftarrow^{\frown} \mathrm{P}\overline{m+1} \\
 & [\mathrm{N}\overline{n+1} \leftarrow g_{a_1}(\mathrm{N}\overline{n+1},\mathrm{N}1,...,\mathrm{N}\overline{n},\mathrm{P}1,...,\mathrm{P}\overline{m},\mathrm{W}\overline{m+2})] \\
 & \mathrm{P}\overline{m+2} \leftarrow \mathrm{P}\overline{m+2}.a_1 \\
 & \mathrm{GOTO\ L}\overline{r+1} \\
 & \vdots \\
 & \\
\mathrm{L}\overline{r} & \mathrm{P}\overline{m+1} \leftarrow^{\frown} \mathrm{P}\overline{m+1} \\
 & [\mathrm{N}\overline{n+1} \leftarrow g_{a_r}(\mathrm{N}\overline{n+1},\mathrm{N}1,...,\mathrm{N}\overline{n},\mathrm{P}1,...,\mathrm{P}\overline{m},\mathrm{P}\overline{m+2})] \\
 & \mathrm{P}m+2 \leftarrow \mathrm{P}m+2.a_r \\
 & \mathrm{GOTO\ L}\overline{r+1} \\
 & \\
\mathrm{L}\overline{r+2} & \mathrm{N1} \leftarrow \mathrm{N}\overline{n+1}
\end{array}
$$

Es fácil chequear que este programa computa h. ■