Combo 6

2 de julio de 2024

1. Lema 11: Efectivamente computable implica efectivamente enumerable

1.1. Enunciado

Si $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-efectivamente computable entonces S es Σ-efectivamente enumerable.

1.2. Demostración

El caso $S = \emptyset$ es trivial. Supongamos $S \neq \emptyset$.

Sea $(\vec{z}, \vec{\gamma}) \in S$, fijo. Sea $\mathbb{P}$ un procedimiento efectivo que compute a $\chi_S^{\omega^n \times \Sigma^{*m}}$. Sea $\mathbb{P}_1$ un procedimiento efectivo que enumere a $\omega^n \times \Sigma^{*m}$. Entonces el siguiente procedimiento $\mathbb{Q}$ enumera a S:

Etapa1 Realizar $\mathbb{P}_1$ con x de entrada para obtener como salida un $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m}$.

Etapa2 Realizar $\mathbb{P}$ con $(\vec{x}, \vec{\alpha})$ de entrada para obtener el valor Booleano e de salida.

Etapa3 Si $e = 1$ dar como dato de salida $(\vec{x}, \vec{\alpha})$. Si $e = 0$ dar como dato de salida $(\vec{z}, \vec{\gamma})$.

Luego, solo nos queda demostrar que $\mathbb{Q}$ enumera a S. Para ello, por lema sabemos que se tiene que cumplir que:

(1) El conjunto de datos de entrada de $\mathbb{Q}$ es ω

(2) $\mathbb{Q}$ se detiene para cada $x \in \omega$

(3) El conjunto de datos de salida de $\mathbb{Q}$ es igual a S. (Es decir, siempre que $\mathbb{Q}$ se detiene, da como salida un elemento de S, y para cada elemento $(\vec{x}, \vec{\alpha}) \in S$, hay un $x \in \omega$ tal que $\mathbb{Q}$ da como salida a $(\vec{x}, \vec{\alpha})$ cuando lo corremos con x como dato de entrada)

Veamos cada punto:

(1) Es trivial de ver.

(2) Es trivial de ver dado que $\mathbb{P}_1$ y $\mathbb{P}$ siempre se detienen.

(3) Para este punto, tenemos que ver los dos casos:

  a) Sea $x \in \omega$, entonces $\mathbb{Q}$ se detiene y da como salida $s \in S$ cuando se corre con x como entrada: Esto se puede comprobar notando que si se devuelve $(\vec{x}, \vec{\alpha})$, entonces $e = 1$, lo que significa que $\chi_S^{\omega^n \times \Sigma^{*m}}((\vec{x}, \vec{a})) = 1$, lo que implica que $(\vec{x}, \vec{\alpha}) \in S$. Caso contrario, se da como salida $(\vec{z}, \vec{\gamma})$, el cual está en S por cómo lo definimos.

  b) Sea $s \in S, \exists x \in \omega$ tal que $\mathbb{Q}$ da como salida a s cuando lo corremos con x como dato de entrada: Como $\mathbb{P}_1$ enumera a $\omega^n \times \Sigma^{*m}$, entonces $\exists y \in \omega$ tal que $\mathbb{P}_1$ se detiene y da como salida s cuando se corre con y como dato de entrada. Luego, es claro notar que si se corre $\mathbb{Q}$ con y como dato de entrada, $e = 1$ y, por lo tanto, se detiene y devuelve s.

Luego, entonces, por Lema tenemos que $\mathbb{Q}$ enumera a S, por lo que S es Σ-efectivamente enumerable. ■

2. Teorema 12 (Caracterización de conjuntos r.e.)

2.1. Enunciado

Dado $S \subseteq \omega^n \times \Sigma^{*m}$, son equivalentes

(1) S es Σ-recursivamente enumerable

(2) $S = I_F$, para alguna $F : D_F \subseteq \omega^k \times \Sigma^{*l} \to \omega^n \times \Sigma^{*m}$ tal que cada $F_{(i)}$ es Σ-recursiva.

(3) $S = D_f$, para alguna funcion Σ-recursiva f

Nota: haga solo la prueba de (2) $\Rightarrow$ (3), caso $k = l = 1$ y $n = m = 2$.

2.2. Demostración de $(2) \Rightarrow (3)$ con $k = l = 1$ y $n = m = 2$

Tenemos que $S \subseteq \omega^2 \times \Sigma^{*2}$ y $F : D_F \subseteq \omega \times \Sigma^* \to \omega^2 \times \Sigma^{*2}$ es tal que $I_F = S$ y $F_{(1)}$, $F_{(2)}$, $F_{(3)}$, $F_{(4)}$ son Σ-recursivas. Para cada $i \in \{1, 2, 3, 4\}$, sea $\mathcal{P}_i$ un programa el cual computa a $F_{(i)}$. Sea $\leq$ un orden total sobre Σ. Definamos

$$H_i = \lambda t x_1 \alpha_1 \left[\neg Halt^{1,1}(t, x_1, \alpha_1, \mathcal{P}_i)\right]$$

Notar que $D_{H_i} = \omega^2 \times \Sigma^*$ y que H_i es Σ-mixta. Ademas sabemos que la funcion $Halt^{1,1}$ es $(\Sigma \cup \Sigma_p)$-p.r. por lo cual resulta facilmente que H_i es $(\Sigma \cup \Sigma_p)$-p.r.. Por la Proposicion de Independencia del Alfabeto tenemos que H_i es Σ-p.r.. Entonces H_i es Σ-computable por lo cual tenemos que hay un macro:

$$[\text{IF } H_i(\text{V2}, \text{V1}, \text{W1}) \text{ GOTO A1}]$$

Para hacer mas intuitivo el uso de este macro lo escribiremos de la siguiente manera

$$\left[\text{IF } \neg Halt^{1,1}(\text{V2}, \text{V1}, \text{W1}, \mathcal{P}_i) \text{ GOTO A1}\right]$$

Para $i = 1, 2$, definamos

$$E_i = \lambda x t x_1 \alpha_1 \left[x \neq E_{\#1}^{1,1}(t, x_1, \alpha_1, \mathcal{P}_i)\right]$$

Para $i = 3, 4$, definamos

$$E_i = \lambda t x_1 \alpha_1 \alpha \left[\alpha \neq E_{*1}^{1,1}(t, x_1, \alpha_1, \mathcal{P}_i)\right]$$

Como estas son iguales a:

$$E_i = \lambda xy\,[x \neq y] \circ \left[p_1^{3,1}, E_{\#1}^{1,1} \circ \left[p_2^{3,1}, p_3^{3,1}, p_4^{3,1}, C_{\mathcal{P}_i}^{3,1}\right]\right]$$

$$E_i = \lambda \alpha\beta\,[\alpha \neq \beta] \circ \left[p_4^{2,2}, E_{*1}^{1,1} \circ \left[p_1^{2,2}, p_2^{2,2}, p_3^{2,2}, C_{\mathcal{P}_i}^{2,2}\right]\right]$$

Y como $E_{\#1}^{1,1}$, $E_{*1}^{1,1}$, $C_{\mathcal{P}_i}^{3,1}$ y $C_{\mathcal{P}_i}^{2,2}$ son $(\Sigma \cup \Sigma_p)$-p.r., tenemos que las funciones E_i son $(\Sigma \cup \Sigma_p)$-p.r.. Ahora, por independencia del alfabeto, tenemos que son Σ-p.r.. Esto significa que son Σ-computables por lo cual para cada $i \in \{1, 2\}$ hay un macro

$$[\text{IF } E_i(\text{V2}, \text{V3}, \text{V1}, \text{W1}) \text{ GOTO A1}]$$

y para cada $i \in \{3, 4\}$ hay un macro

$$[\text{IF } E_i(\text{V2}, \text{V1}, \text{W1}, \text{W2}) \text{ GOTO A1}]$$

Haremos mas intuitiva la forma de escribir estos macros, por ejemplo para $i = 1$, lo escribiremos de la siguiente manera

$$\left[\text{IF V2} \neq E_{\#1}^{1,1}(\text{V3}, \text{V1}, \text{W1}, \mathcal{P}_1) \text{ GOTO A1}\right]$$

Ya que la funcion $f = \lambda x[(x)_1]$ es Σ-p.r., ella es Σ-computable por lo cual hay un macro

$$[\text{V2} \leftarrow f(\text{V1})]$$

el cual escribiremos de la siguiente manera:

$$[\text{V2} \leftarrow (\text{V1})_1]$$

Similarmente hay macros:

$$[\text{W1} \leftarrow *^{\leq}(\text{V1})_3]$$

$$[\text{V2} \leftarrow (\text{V1})_2]$$

Sea $\mathcal{P}$ el siguiente programa de $\mathcal{S}^{\Sigma}$:

$$\begin{array}{l}
\text{L1 N20} \leftarrow \text{N20} + 1 \\
[\text{N10} \leftarrow (\text{N20})_1] \\
[\text{N3} \leftarrow (\text{N20})_2] \\
[\text{P3} \leftarrow *^{\leq}(\text{N20})_3] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_1) \text{ GOTO L1}\right] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_2) \text{ GOTO L1}\right] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_3) \text{ GOTO L1}\right] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_4) \text{ GOTO L1}\right] \\
\left[\text{IF N1} \neq E^{1,1}_{\#1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_1) \text{ GOTO L1}\right] \\
\left[\text{IF N2} \neq E^{1,1}_{\#1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_2) \text{ GOTO L1}\right] \\
\left[\text{IF P1} \neq E^{1,1}_{*1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_3) \text{ GOTO L1}\right] \\
\left[\text{IF P2} \neq E^{1,1}_{*1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_4) \text{ GOTO L1}\right]
\end{array}$$

Tenemos que demostrar que este programa computa a $p_1^{2,2}|_S$, lo que es equivalente a demostrar que $p_1^{2,2}|_S = \Psi_{\mathcal{P}}^{2,2,\#}$, Para ello, veamos los casos:

- $D_{p_1^{2,2}|_S} = D_{\Psi_{\mathcal{P}}^{2,2,\#}}$: Es fácil de ver notando que $D_{p_1^{2,2}|_S} = S$ por la clausura y $D_{\Psi_{\mathcal{P}}^{2,2,\#}} = S$ dado que, en otro caso, no se detiene porque $\mathcal{P}_i$ computa $F_{(i)}$ y siempre algún predicado daría 1 (Halt).

- $\forall (x, y, \alpha, \beta) \in S, p_1^{2,2}|_S((x, y, \alpha, \beta)) = \Psi_{\mathcal{P}}^{2,2,\#}((x, y, \alpha, \beta))$: Claramente $p_1^{2,2}|_S((x, y, \alpha, \beta)) = x$, por lo que se reduce a comprobar que $\Psi_{\mathcal{P}}^{2,2,\#}((x, y, \alpha, \beta)) = x$. Para ello, nos concentremos en el programa $\mathcal{P}$. Como $(x, y, \alpha, \beta) \in S$, entonces sabemos que $\exists k \in N : F((k)_1, *^{\leq}(k)_2) = (x, y, \alpha, \beta)$, Digamos, ahora, que con p pasos todos los programas $\mathcal{P}_i$ se detuvieron para el estado inicial $||(k)_1, *^{\leq}(k)_2||$, entonces se puede notar que $\mathcal{P}$ se detiene y el valor almacenado en N1 será x para el estado inicial $||2^p 3^{(k)_1} 5^{(k)_2}||$.

Luego, se demuestra que $\mathcal{P}$ computa a $p_1^{2,2}|_S$, por lo que es Σ-computable. Esto implica que es Σ-recursiva, lo cual prueba (3) dado que $Dom(p_1^{2,2}|_S) = S$. ■