Lenguajes Formales y Computabilidad

Teoremas: Combo 8

Tomás Maraschio

June 26, 2025

Lema. *Supongamos $\Sigma_p \subseteq \Sigma$. Entonces $AutoHalt^{\Sigma}$ no es Σ-recursivo.*

Proof.

$$AutoHalt^{\Sigma} = \lambda\mathcal{P}[(\exists t \in \omega) Halt^{0,1}(t, \mathcal{P}, \mathcal{P})]$$

Notar que $D_{AutoHalt^{\Sigma}} = Pro^{\Sigma}$ y $\forall\mathcal{P} \in Pro^{\Sigma}$ se cumple

$$(*)\ AutoHalt^{\Sigma}(\mathcal{P}) = 1 \iff \mathcal{P} \text{ se detiene partiendo de } \|\mathcal{P}\|$$

Supongamos que $AutoHalt^{\Sigma}$ es Σ-recursiva, entonces, por el 2° manantial de macros, existe el macro

$$[\text{ IF } AutoHalt^{\Sigma}(\text{W1}) \text{ GOTO A1}]$$

Luego, sea $\mathcal{P}_0 = \text{L1}[\text{ IF } AutoHalt^{\Sigma}(\text{P1}) \text{ GOTO L1}]$, tenemos que $\mathcal{P}_0$ se detiene partiendo de $\|\mathcal{P}_0\|$ si y solo si $AutoHalt^{\Sigma}(\mathcal{P}_0) = 0$. Pero esto contradice la propiedad (*), y viene de suponer que $AutoHalt^{\Sigma}$ es Σ-recursiva, por lo que llegamos a que $AutoHalt^{\Sigma}$ no puede ser Σ-recursiva. ■

Teorema. *Supongamos $\Sigma_p \subseteq \Sigma$. Entonces $AutoHalt^{\Sigma}$ no es Σ-efectivamente computable. Es decir, no hay ningun procedimiento efectivo que decida si un programa de S^{Σ} termina partiendo de sí mismo.*

Proof. La Tesis de Church dice que toda función Σ-efectivamente computable es Σ-recursiva. Por lo que si $AutoHalt^{\Sigma}$ fuese Σ-efectivamente computable, esta sería también Σ-recursiva, contradiciendo el lema anterior. Luego, $AutoHalt^{\Sigma}$ no es Σ-efectivamente computable. ■

Lema. *Supongamos $\Sigma_p \subseteq \Sigma$. Entonces*

$$A = \{\mathcal{P} \in Pro^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 1\}$$

es Σ-recursivamente enumerable y no es Σ-recursivo. Mas aún, el conjunto

$$N = \{\mathcal{P} \in Pro^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 0\}$$

no es Σ-recursivamente enumerable.

Proof. Sea $P = \lambda t\mathcal{P}[Halt^{0,1}(t,\mathcal{P},\mathcal{P})] = Halt^{0,1} \circ [p_1^{1,1}, p_2^{1,1}, p_2^{1,1}]$. Notar que $D_P = \omega \times Pro^{\Sigma}$. Además, P es $(\Sigma \cup \Sigma_p)$-p.r., que en este caso es lo mismo que Σ-p.r. Entonces, $M(P)$ es Σ-recursiva y además

$$D_{M(P)} = \{\mathcal{P} \in Pro^{\Sigma} : (\exists t \in \omega) P(t,\mathcal{P}) = 1\} = \{\mathcal{P} \in Pro^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 1\} = A$$

Luego, por el teorema de caracterización de conjuntos Σ-recursivamente enumerables, que entre otras cosas dice que un conjunto es Σ-r.e. si y solo si es el dominio de alguna función Σ-recursiva, tenemos que $\underline{A \text{ es } \Sigma\text{-r.e.}}$

Supongamos ahora que N es Σ-r.e. Entonces, por el lema de restricción de funciones recursiva tenemos que $C_0^{0,1}|_N$ es Σ-recursiva. Por el mismo lema, y como sabemos que A es Σ-r.e., $C_1^{0,1}|_A$ es Σ-recursiva. Además, notar que $A \cup N = Pro^{\Sigma}$ y $A \cap N = \emptyset$. Entonces, por el lema de división por casos de funciones Σ-recursivas, tenemos que la función

$$AutoHalt^{\Sigma} = C_1^{0,1}|_A \cup C_0^{0,1}|_N$$

es Σ-recursiva. Pero esto contradice el lema probado anteriormente, por lo que $\underline{N \text{ no puede ser } \Sigma\text{-r.e.}}$

Por último, supongamos que A es Σ-recursivo, entonces el conjunto

$$N = Pro^{\Sigma} - A$$

también es Σ-recursivo. Pero esto es un absurdo porque N no es Σ-r.e. y por lo tanto tampoco es Σ-recursivo. Así, $\underline{A \text{ no es } \Sigma\text{-recursivo.}}$ ■

Teorema. (Neumann vence a Godel). *Si h es Σ-recursiva, entonces h es Σ-computable. (En la induccion de la prueba hacer solo el caso $h = M(P)$)*

Proof. Probaré por inducción sobre k que si $h \in R_k^{\Sigma}$ entonces h es Σ-computable.

Caso base. Para cada una de las funciones de R_0^{Σ} daré un programa que las compute.

Suc:

$$\text{N1} \leftarrow \text{N1} + 1$$

Pred:

```
L1  IF N1 ≠ 0 GOTO L2
    GOTO L1
L2  N1 ← N1∸1
```

$C_0^{0,0}$ y $C_\varepsilon^{0,0}$:

$$\text{SKIP}$$

d_a, para todo $a \in \Sigma$:

$$\text{P1} \leftarrow \text{P1}.a$$

$p_j^{n,m}$, para todo $n, m \in \omega$ y $1 \leq j \leq n$:

$$\text{N1} \leftarrow \text{N}\overline{j}$$

$p_j^{n,m}$, para todo $n, m \in \omega$ y $n < j \leq m$:

$$\text{P1} \leftarrow \text{P}\overline{j-n}$$

<u>Caso inductivo.</u> Supongamos que la hipótesis vale para $k \in \omega$. Sea $h = M(P) \in R_{k+1}^{\Sigma}$ tal que $h : D_h \subseteq \omega^{n+1} \times \Sigma^{*m} \to \omega$ y $P : \omega^n \times \Sigma^{*m} \to \omega$ un predicado en R_k^{Σ}. Por hipótesis inductiva, P es Σ-computable, entonces por 1° manantial de macros existe el macro

$$[\text{IF P(V}\overline{n+1}\text{, V1, ..., V}\overline{n}\text{, W1, ..., W}\overline{m}\text{) GOTO A1}]$$

Así, el siguiente programa computa a $M(P)$:

L2	[IF P(N$\overline{n+1}$, N1, ..., N$\overline{n}$, P1, ..., P$\overline{m}$) GOTO L1]
	N$\overline{n+1}$ $\leftarrow$ N$\overline{n+1}$ + 1
	GOTO L2
L1	N1 $\leftarrow$ N$\overline{n+1}$

Esto es porque, para $(x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_m) \in \omega^n \times \Sigma^{*m}$, si el programa se detiene partiendo de $\|x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_m\|$ entonces en N1 quedará guardado el primer t tal que $P(t, x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_m) = 1$. Si no se detiene partiendo de $\|x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_m\|$ es porque no existe t tal que $P(t, x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_m) = 1$, lo que significa que $(x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_m) \notin D_h$. ∎