Lenguajes Formales y Computabilidad

Teoremas: Combo 9

Nicolás Cagliero

July 3, 2025

Lema (Lema de división por casos para funciones Σ-recursivas).
Supongamos $f_i : D_{f_i} \subseteq \omega^n \times \Sigma^{*m} \to O$, $i = 1, \ldots, k$, son funciones Σ-recursivas tales que $D_{f_i} \cap D_{f_j} = \emptyset$ para $i \neq j$. Entonces la función $f_1 \cup \ldots \cup f_k$ es Σ-recursiva. (Haga el caso $k = 2$, $n = m = 1$ y $O = \omega$)

Dem Sean $\mathcal{P}_1$ y $\mathcal{P}_2$ programas que computen las funciones f_1 y f_2, respectivamente. Para $i = 1, 2$, definamos

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

Notar que $D_{H_i} = \omega^2 \times \Sigma^*$ y que H_i es Σ-mixta. Además sabemos que la función $Halt^{1,1}$ es $(\Sigma \cup \Sigma_p)$-p.r., por lo cual resulta fácilmente que H_i es $(\Sigma \cup \Sigma_p)$-p.r.. Por el Teorema de Independencia del Alfabeto tenemos que H_i es Σ-p.r.. y por lo tanto el Segundo Manantial de Macros nos dice que en $\mathcal{S}^\Sigma$ hay un macro:

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

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

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

Ya que cada f_i es Σ-recursiva, el Primer Manantial nos dice que en $\mathcal{S}^\Sigma$ hay macros

$$[\text{V2} \leftarrow f_1(\text{V1}, \text{W1})]$$
$$[\text{V2} \leftarrow f_2(\text{V1}, \text{W1})]$$

Sea $\mathcal{P}$ el siguiente programa:

```
L1  N20 ← N20 + 1
    [IF Halt^{1,1}(N20, N1, P1, 𝒫_1) GOTO L2]
    [IF Halt^{1,1}(N20, N1, P1, 𝒫_2) GOTO L3]
    GOTO L1
L2  [N1 ← f_1(N1, P1)]
    GOTO L4
L3  [N1 ← f_2(N1, P1)]
L4  SKIP
```

Nótese que $\mathcal{P}$ computa la función $f_1 \cup f_2$ pues primero nos fijamos que el contenido en N1 y P1 sea parte del dominio de algunas de las dos funciones. En caso que no sea de ninguno de los dominios, el programa no termina. Cuando vemos que en efecto pertenece a alguno de los dos dominios, dejamos en N1 el valor correspondiente al resultado de aplicar f_i con esos valores. ■

Teorema (Gödel vence a Neumann). *Si* $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$ *es* Σ*-computable, entonces* f *es* Σ*-recursiva.*

Dem Sea $\mathcal{P}_0$ un programa que compute a f. Primero veremos que f es $(\Sigma \cup \Sigma_p)$-recursiva. Note que

$$f = E_{\#1}^{n,m} \circ [T^{n,m} \circ [p_1^{n,m}, \ldots, p_{n+m}^{n,m}, C_{\mathcal{P}_0}^{n,m}], p_1^{n,m}, \ldots, p_{n+m}^{n,m}, C_{\mathcal{P}_0}^{n,m}]$$

donde cabe destacar que $p_1^{n,m}, \ldots, p_{n+m}^{n,m}$ son las proyecciones respecto del alfabeto $\Sigma \cup \Sigma_p$, es decir que tienen dominio $\omega^n \times (\Sigma \cup \Sigma_p)^{*m}$. Esto nos dice que f es $(\Sigma \cup \Sigma_p)$-recursiva. O sea que el Teorema de Independencia del Alfabeto nos dice que f es Σ-recursiva. ■