

Definiciones y convenciones notacionales

Combo 1

Defina:

1. Cuándo un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ -recursivo (no hace falta que defina "función Σ -recursiva")
2. $\langle s_1, s_2, \dots \rangle$
3. " f es una función Σ -mixta"
4. "familia Σ -indexada de funciones"
5. $R(f, \mathcal{G})$

Resolución

1. Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ -recursivo si su función característica $\chi_S^{\omega^n \times \Sigma^{*m}}$ es Σ -recursiva.
2. Dada una infinitupla $(s_1, s_2, \dots) \in \omega^{[N]}$, se usa $\langle s_1, s_2, \dots \rangle$ para denotar al número $x = \prod_{i=1}^{\infty} pr(i)^{s_i}$
3. Sea Σ un alfabeto finito y sea f una función, diremos que es Σ -mixta si $\exists n, m \geq 0 : D_f \subseteq \omega^n \times \Sigma^{*m}$ e $I_f \subseteq O$ donde $O \in \{\omega, \Sigma^*\}$
4. Dado un alfabeto Σ , una familia Σ -indexada de funciones es una función \mathcal{G} tal que $D_{\mathcal{G}} = \Sigma$ y $\forall a \in D_{\mathcal{G}}, \mathcal{G}(a)$ es una función.
5. La recursión primitiva para el caso de *variable alfabética* se define de forma distinta para los casos de *valores numéricos* o *alfabéticos*. Por ello, veamos cada uno:
 - **Valores numéricos:** Sea Σ un alfabeto finito, y sean f una función y \mathcal{G} una familia Σ -indexada de funciones tales que:

$$f : S_1 \times \dots \times S_n \times L_1 \times \dots \times L_m \rightarrow \omega$$

$$\mathcal{G}_a : \omega \times S_1 \times \dots \times S_n \times L_1 \times \dots \times L_m \times \Sigma^* \rightarrow \omega$$

para cada $a \in \Sigma$, y con $S_i \subseteq \omega$ y $L_i \subseteq \Sigma^*$ conjuntos no vacíos, entonces definimos

$$R(f, \mathcal{G}) : S_1 \times \dots \times S_n \times L_1 \times \dots \times L_m \times \Sigma^* \rightarrow \omega$$

$$R(f, \mathcal{G})(\vec{x}, \vec{\alpha}, \varepsilon) = f(\vec{x}, \vec{\alpha})$$

$$R(f, \mathcal{G})(\vec{x}, \vec{\alpha}, \alpha a) = \mathcal{G}_a(R(f, \mathcal{G})(\vec{x}, \vec{\alpha}, \alpha), \vec{x}, \vec{\alpha}, \alpha)$$

y decimos que $R(f, \mathcal{G})$ es obtenida por recursión primitiva a partir de f y \mathcal{G} .

- **Valores alfabéticos:** Sea Σ un alfabeto finito, y sean f una función y \mathcal{G} una familia Σ -indexada de funciones tales que:

$$f : S_1 \times \dots \times S_n \times L_1 \times \dots \times L_m \rightarrow \Sigma^*$$

$$\mathcal{G}_a : S_1 \times \dots \times S_n \times L_1 \times \dots \times L_m \times \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$$

para cada $a \in \Sigma$, y con $S_i \subseteq \omega$ y $L_i \subseteq \Sigma^*$ conjuntos no vacíos, entonces definimos

$$R(f, \mathcal{G}) : S_1 \times \dots \times S_n \times L_1 \times \dots \times L_m \times \Sigma^* \rightarrow \Sigma^*$$

$$R(f, \mathcal{G})(\vec{x}, \vec{\alpha}, \varepsilon) = f(\vec{x}, \vec{\alpha})$$

$$R(f, \mathcal{G})(\vec{x}, \vec{\alpha}, \alpha a) = \mathcal{G}_a(\vec{x}, \vec{\alpha}, \alpha, R(f, \mathcal{G})(\vec{x}, \vec{\alpha}, \alpha))$$

y decimos que $R(f, \mathcal{G})$ es obtenida por recursión primitiva a partir de f y \mathcal{G} .

Combo 2

Defina:

1. $d \vdash^n d'$ (no hace falta que defina \vdash)
2. $L(M)$
3. $H(M)$
4. " f es una función de tipo (n, m, s) "
5. (x)
6. $(x)_i$

Resolución

1. Para $d, d' \in Des, n \geq 0$, escribiremos $d \vdash^n d'$ si $\exists d_1, \dots, d_{n+1}$ tales que $d = d_1, d' = d_{n+1}$ y $d_i \vdash d_{i+1} \forall i = 1, \dots, n$.
 - *Notar que $d \vdash^0 d' \Leftrightarrow d = d'$*
2. Diremos que una palabra $\alpha \in \Sigma^*$ es aceptada por M por *alcance de estado final* cuando $\lfloor q_0 B \alpha \rfloor \vdash^* d$, con $d : St(d) \in F$. Luego, el *lenguaje aceptado por M por alcance de estado final* se define como

$$L(M) = \{\alpha \in \Sigma^* : \alpha \text{ es aceptada por } M \text{ por alcance de estado final}\}$$
3. Diremos que una palabra $\alpha \in \Sigma^*$ es aceptada por M por *detención* cuando M se detiene partiendo de $\lfloor q_0 B \alpha \rfloor$. Luego, el *lenguaje aceptado por M por detención* se define como $H(M) = \{\alpha \in \Sigma^* : \alpha \text{ es aceptada por } M \text{ por detención}\}$
4. Si f es una función Σ -mixta y $n, m \in \omega : D_f \subseteq \omega^n \times \Sigma^{*m}$,
 - Si $I_f \subseteq \omega$, decimos que f es de tipo $(n, m, \#)$
 - Si $I_f \subseteq \Sigma^*$, decimos que f es de tipo $(n, m, *)$
5. Dado $x \in N$, usaremos (x) para denotar a la única infinitupla $(s_1, s_2, \dots) \in \omega^{[N]}$ tal que

$$x = \langle s_1, s_2, \dots \rangle = \prod_{i=1}^{\infty} pr(i)^{s_i}$$

6. Para cada $i \in N$, usaremos $(x)_i$ para denotar a s_i de la anterior infinitupla. Es decir, $(x)_i$ es el exponente de $pr(i)$ en la única factorización prima de x

Combo 3

Defina:

1. Cuándo un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ -recursivamente enumerable (no hace falta que defina "función Σ -recursiva")
2. s^{\leq}
3. $*^{\leq}$
4. $\#^{\leq}$

Resolución

1. Diremos que un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ -recursivamente enumerable cuando sea vacío o haya una función $F : \omega \rightarrow \omega^n \times \Sigma^{*m}$ tal que $I_F = S$ y $F_{(i)}$ sea Σ -recursiva $\forall i \in \{1, \dots, n+m\}$

Los siguientes puntos se definen en base a Σ alfabeto no vacío y \leq orden total sobre Σ , siendo $\Sigma = \{a_1, \dots, a_n\}$ con $a_1 < a_2 < \dots < a_n$. Luego:

2. La función *siguiente* se define como

$$s^{\leq} : \Sigma^* \rightarrow \Sigma^*$$

$$s^{\leq}((a_n)^m) = (a_1)^{m+1} \forall m \geq 0$$

$$s^{\leq}(\alpha a_i (a_n)^m) = \alpha a_{i+1} (a_1)^m \forall \alpha \in \Sigma^*, i \in \{1, \dots, n-1\}, m \geq 0$$

3. Función que asigna a cada $n \in \omega$ la $n+1$ -ésima palabra de la lista:

$$*^{\leq} : \omega \rightarrow \Sigma^*$$

$$*^{\leq}(0) = \varepsilon$$

$$*^{\leq}(n+1) = s^{\leq}(*^{\leq}(n))$$

4. Inversa de la anterior:

$$\#^{\leq} : \Sigma^* \rightarrow \omega$$

$$\#^{\leq}(\varepsilon) = 0$$

$$\#^{\leq}(a_{i_k} \dots a_{i_0}) = \sum_{j=0}^k i_j n^j$$

Combo 4

Defina cuándo una función $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$ es llamada Σ -efectivamente computable y defina "el procedimiento \mathbb{P} computa a la función f ".

Resolución

Una función Σ -mixta $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ (para $O \in \{\omega, \Sigma^*\}$) es Σ -efectivamente computable si hay un procedimiento \mathbb{P} tal que:

- El conjunto de datos de entrada de \mathbb{P} es $\omega^n \times \Sigma^{*m}$
- El conjunto de datos de salida está contenido en O
- Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces \mathbb{P} se detiene partiendo de $(\vec{x}, \vec{\alpha})$ y da como salida $f(\vec{x}, \vec{\alpha})$
- Si $(\vec{x}, \vec{\alpha}) \notin D_f$, entonces \mathbb{P} no se detiene partiendo de $(\vec{x}, \vec{\alpha})$

En estos casos, diremos que este \mathbb{P} computa a la función f .

Combo 5

Defina cuándo un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ -efectivamente computable y defina "el procedimiento efectivo \mathbb{P} decide la pertenencia a S ".

Resolución

Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ -efectivamente computable cuando la función $\chi_S^{\omega^n \times \Sigma^{*m}}$ es Σ -efectivamente computable.

Es decir, S es Σ -efectivamente computable si existe un procedimiento \mathbb{P} tal que:

- 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} se detiene partiendo de $(\vec{x}, \vec{\alpha})$ y da como salida 1 si $(\vec{x}, \vec{\alpha}) \in S$ y 0 en caso contrario.

En este caso, decimos que el procedimiento efectivo \mathbb{P} decide la pertenencia a S .

Combo 6

Defina cuándo un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ -recursivamente enumerable y defina "el procedimiento efectivo \mathbb{P} enumera a S ".

Resolución

Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ -efectivamente enumerable si es vacío o $\exists F : \omega \rightarrow \omega^n \times \Sigma^{*m}$ tal que $I_F = S$ y $F_{(i)}$ es Σ -efectivamente computable $\forall i \in \{1, \dots, n+m\}$.

Es decir, $S \neq \emptyset$ es Σ -efectivamente enumerable si existe un procedimiento efectivo \mathbb{P} tal que:

- El conjunto de datos de entrada de \mathbb{P} es ω
- \mathbb{P} se detiene para cada $x \in \omega$
- El conjunto de datos de salida de \mathbb{P} es igual a S

En este caso, decimos que el procedimiento efectivo \mathbb{P} enumera a S .

Combo 7

Defina cuándo una función $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$ es llamada Σ -Turing computable y defina "la máquina de Turing M computa a la función f ".

Resolución

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

- Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces $\exists p \in Q : \lfloor q_0 B \vdash^{x_1} B \dots B \vdash^{x_n} B \alpha_1 B \dots B \alpha_m \rfloor \vdash^* \lfloor p B \vdash^{f(\vec{x}, \vec{\alpha})} \rfloor$ y $\lfloor p B \vdash^{f(\vec{x}, \vec{\alpha})} \rfloor \not\vdash d \forall d \in Des$
- Si $(\vec{x}, \vec{\alpha}) \in (\omega^n \times \Sigma^{*m}) - D_f$, entonces M **no** se detiene partiendo de $\lfloor q_0 B \vdash^{x_1} B \dots B \vdash^{x_n} B \alpha_1 B \dots B \alpha_m \rfloor$

En este caso, diremos que la máquina de Turing M computa a la función f .

Combo 8

Defina:

1. $M(P)$
2. Lt
3. Conjunto rectangular
4. " S es un conjunto de tipo (n, m) "

Resolución

1. Este caso se trata de **minimización de variable numérica**. Sea Σ un alfabeto finito y sea $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \rightarrow \omega$ un predicado, dado $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m}$, cuando exista al menos un $t \in \omega : P(t, \vec{x}, \vec{\alpha}) = 1$, usaremos $\min_t P(t, \vec{x}, \vec{\alpha})$ para denotar al menor de tales t 's. Con ello, definimos

$$M(P) = \lambda \vec{x} \vec{\alpha} \left[\min_t P(t, \vec{x}, \vec{\alpha}) \right]$$

El cual cumple que:

$$D_{M(P)} = \{(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} : (\exists t \in \omega) P(t, \vec{x}, \vec{\alpha})\}$$

$$M(P)(\vec{x}, \vec{\alpha}) = \min_t P(t, \vec{x}, \vec{\alpha}), \forall (\vec{x}, \vec{\alpha}) \in D_{M(P)}$$

Y diremos que $M(P)$ se obtiene por *minimización de variable numérica* a partir de P .

2. Definimos la función del *mayor factor primo* como

$$Lt : N \rightarrow \omega$$

$$Lt(x) = \begin{cases} \max\{i \in N : (x)_i \neq 0\} & \text{si } x \neq 1 \\ 0 & \text{si } x = 1 \end{cases}$$

3. Sea Σ un alfabeto finito, un conjunto Σ -mixto S es llamado *rectangular* si es de la forma $S_1 \times \dots \times S_n \times L_1 \times \dots \times L_m$ con $S_i \subseteq \omega$ y $L_j \subseteq \Sigma^*$
4. Dado un conjunto Σ -mixto S y $n, m \in \omega : S \subseteq \omega^n \times \Sigma^{*m}$, entonces S es de tipo (n, m)

Combo 9

Defina:

1. " I es una instrucción de S^Σ "
2. " \mathcal{P} es un programa de S^Σ "
3. $I_i^{\mathcal{P}}$
4. $n(\mathcal{P})$
5. Bas

Resolución

1. Una *instrucción de S^Σ* es ya sea una instrucción básica de S^Σ , o una palabra de la forma αI , donde $\alpha \in \{L\bar{n} : n \in N\}$ e I es una instrucción básica de S^Σ .
 - Cuando I es de la forma $L\bar{n}J$ con J una instrucción básica, diremos que $L\bar{n}$ es la *label de I*
 - Una *instrucción básica de S^Σ* es una palabra $(\Sigma \cup \Sigma_P)^*$ la cual es de alguna de las siguientes formas (donde $a \in \Sigma$; $k, n \in N$):
 - $N\bar{k} \leftarrow N\bar{k} + 1$

- $N\bar{k} \leftarrow N\bar{k} - 1$
 - $N\bar{k} \leftarrow N\bar{n}$
 - $N\bar{k} \leftarrow 0$
 - $P\bar{k} \leftarrow P\bar{k}.a$
 - $P\bar{k} \leftarrow \neg P\bar{k}$
 - $P\bar{k} \leftarrow P\bar{n}$
 - $P\bar{k} \leftarrow \varepsilon$
 - IF $N\bar{k} \neq 0$ GOTO $L\bar{n}$
 - IF $P\bar{k}$ BEGINS a GOTO $L\bar{n}$
 - GOTO $L\bar{n}$
 - SKIP
2. Un *programa* de S^Σ es una palabra de la forma $I_1 I_2 \dots I_n$ donde $n \geq 1, I_1, \dots, I_n \in Ins^\Sigma$ y además se cumple la *ley de los GOTO*: $\forall i \in \{1, \dots, n\}$, si $GOTO L\bar{m}$ es un tramo final de I_i , entonces $\exists j \in \{1, \dots, n\}$ tal que I_j tiene label $L\bar{m}$
 3. Definimos $I_i^{\mathcal{P}}$ como la i -ésima instrucción de \mathcal{P} y, además, $I_i^{\mathcal{P}} = \varepsilon$ cuando $i = 0$ o $i > n(\mathcal{P})$
 4. Definimos $n(\mathcal{P})$ como la cantidad de instrucciones de \mathcal{P}
 5. Definimos $Bas : Ins^\Sigma \rightarrow (\Sigma \cup \Sigma_p)^*$ dada por

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

Combo 10

Defina, relativo al lenguaje S^Σ :

1. "Estado"
2. "Descripción instantánea"
3. $S_{\mathcal{P}}$
4. "Estado obtenido luego de t pasos, partiendo del estado $(\vec{s}, \vec{\sigma})$ "
5. " \mathcal{P} se detiene (luego de t pasos), partiendo del estado $(\vec{s}, \vec{\sigma})$ "

Resolución

1. Un estado es un par $(\vec{s}, \vec{\sigma}) = ((s_1, s_2, \dots), (\sigma_1, \sigma_2, \dots)) \in \omega^{[N]} \times \Sigma^{*[N]}$ y, si $i \geq 1$, entonces diremos que s_i es el contenido o valor de la variable $N\bar{i}$ en el estado $(\vec{s}, \vec{\sigma})$ y σ_i es el contenido o valor de la variable $P\bar{i}$ en el estado $(\vec{s}, \vec{\sigma})$
2. Una descripción instantánea es una terna $(i, \vec{s}, \vec{\sigma})$ tal que $(\vec{s}, \vec{\sigma})$ es un estado e $i \in \omega$. Intuitivamente, $(i, \vec{s}, \vec{\sigma})$ nos dice que las variables están en el estado $(\vec{s}, \vec{\sigma})$ y que la instrucción que debemos realizar es $I_i^{\mathcal{P}}$

3. Dado un programa \mathcal{P} , definimos $S_{\mathcal{P}} : \omega \times \omega^{[N]} \times \Sigma^{*[N]} \rightarrow \omega \times \omega^{[N]} \times \Sigma^{*[N]}$ como la función que asignará a una descripción instantánea $(i, \vec{s}, \vec{\sigma})$ la descripción instantánea sucesora de $(i, \vec{s}, \vec{\sigma})$ con respecto a \mathcal{P} . Es decir, hay varios casos posibles:
- Si $i \notin \{1, \dots, n(\mathcal{P})\}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i, \vec{s}, \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow N\bar{k} - 1$, entonces
 $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, (s_1, \dots, s_{k-1}, s_{\bar{k}} - 1, s_{k+1}, \dots), \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow N\bar{k} + 1$, entonces
 $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, (s_1, \dots, s_{k-1}, s_{\bar{k}} + 1, s_{k+1}, \dots), \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow N\bar{n}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, (s_1, \dots, s_{k-1}, s_{\bar{n}}, s_{k+1}, \dots), \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow 0$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, (s_1, \dots, s_{k-1}, 0, s_{k+1}, \dots), \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = \text{IF } N\bar{k} \neq 0 \text{ GOTO } L\bar{m}$, entonces hay dos posibilidades:
 - Si el valor contenido en $N\bar{k}$ es 0, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, \vec{s}, \vec{\sigma})$
 - Si el valor contenido en $N\bar{k}$ es no nulo, entonces
 $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (\min\{l : I_l^{\mathcal{P}} \text{ tiene label } L\bar{m}\}, \vec{s}, \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow \neg P\bar{k}$, entonces
 $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, \vec{s}, (\sigma_1, \dots, \sigma_{k-1}, \neg \sigma_k, \sigma_{k+1}, \dots))$
 - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow P\bar{k}.a$, entonces
 $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, \vec{s}, (\sigma_1, \dots, \sigma_{k-1}, \sigma_k a, \sigma_{k+1}, \dots))$
 - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow P\bar{n}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, \vec{s}, (\sigma_1, \dots, \sigma_{k-1}, \sigma_{\bar{n}}, \sigma_{k+1}, \dots))$
 - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow \varepsilon$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, \vec{s}, (\sigma_1, \dots, \sigma_{k-1}, \varepsilon, \sigma_{k+1}, \dots))$
 - Si $Bas(I_i^{\mathcal{P}}) = \text{IF } P\bar{k} \text{ BEGINS } a \text{ GOTO } L\bar{m}$, entonces hay dos posibilidades:
 - Si la palabra contenida en $P\bar{k}$ comienza con a , entonces
 $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (\min\{l : I_l^{\mathcal{P}} \text{ tiene label } L\bar{m}\}, \vec{s}, \vec{\sigma})$
 - Si la palabra contenida en $P\bar{k}$ no comienza con a , entonces
 $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, \vec{s}, \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = \text{GOTO } L\bar{m}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (\min\{l : I_l^{\mathcal{P}} \text{ tiene label } L\bar{m}\}, \vec{s}, \vec{\sigma})$
 - Si $Bas(I_i^{\mathcal{P}}) = \text{SKIP}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i + 1, \vec{s}, \vec{\sigma})$
4. Diremos que $S_{\mathcal{P}}(S_{\mathcal{P}}(\dots(S_{\mathcal{P}}(1, \vec{s}, \vec{\sigma})).\dots)) = (j, \vec{u}, \vec{\eta})$ con $S_{\mathcal{P}}$ aplicado t veces, es la *descripción instantánea obtenida luego de t pasos partiendo del estado $(\vec{s}, \vec{\sigma})$* , y $(\vec{u}, \vec{\eta})$ es el estado obtenido luego de t pasos partiendo del estado $(\vec{s}, \vec{\sigma})$
5. Cuando la primer coordenada de $S_{\mathcal{P}}(S_{\mathcal{P}}(\dots(S_{\mathcal{P}}(1, \vec{s}, \vec{\sigma})).\dots))$ (con $S_{\mathcal{P}}$ aplicado t veces) es $n(\mathcal{P}) + 1$, diremos que \mathcal{P} se detiene (luego de t pasos), partiendo desde el estado $(\vec{s}, \vec{\sigma})$

Combo 11

Defina:

1. $\Psi_{\mathcal{P}}^{n,m,\#}$
2. " f es Σ -computable"
3. " \mathcal{P} computa a f "

4. $M^{\leq}(P)$

Resolución

1. Dado $\mathcal{P} \in Pro^{\Sigma}$, definimos $\Psi_{\mathcal{P}}^{n,m,\#}$ como:

$$D_{\Psi_{\mathcal{P}}^{n,m,\#}} = \{(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} : \mathcal{P} \text{ termina partiendo de } ||\vec{x}, \vec{\alpha}||\}$$

$$\Psi_{\mathcal{P}}^{n,m,\#}(\vec{x}, \vec{\alpha}) = \text{valor de N1 cuando } \mathcal{P} \text{ termina partiendo de } ||\vec{x}, \vec{\alpha}||$$

2. Una función Σ -mixta $f : S \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$ es Σ -computable si existe un programa $\mathcal{P} \in \mathcal{S}^{\Sigma}$ tal que $f = \Psi_{\mathcal{P}}^{n,m,\#}$

- Se define de forma análoga para funciones Σ -mixtas $f : S \subseteq \omega^n \times \Sigma^{*m} \rightarrow \Sigma^*$ con $f = \Psi_{\mathcal{P}}^{n,m,*}$

3. En el caso anterior, decimos que f es *computada* por \mathcal{P}

4. Sea $\Sigma \neq \emptyset$ un alfabeto con \leq un orden total sobre este, y sea

$P : D_P \subseteq \omega^n \times \Sigma^{*m} \times \Sigma^* \rightarrow \omega$ un predicado, dado $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m}$, cuando exista al menos un $\alpha \in \Sigma^*$ tal que $P(\vec{x}, \vec{\alpha}, \alpha) = 1$, usaremos $\min_{\alpha}^{\leq} P(\vec{x}, \vec{\alpha}, \alpha)$ para denotar al menor de tales α 's. Con ello, definimos:

$$M^{\leq}(P) = \lambda \vec{x} \vec{\alpha} [\min_{\alpha}^{\leq} P(\vec{x}, \vec{\alpha}, \alpha)]$$

El cual cumple que:

$$D_{M^{\leq}(P)} = \{(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} : (\exists \alpha \in \Sigma^*) P(\vec{x}, \vec{\alpha}, \alpha) = 1\}$$

$$M^{\leq}(P)(\vec{x}, \vec{\alpha}) = \min_{\alpha}^{\leq} P(\vec{x}, \vec{\alpha}, \alpha), \forall (\vec{x}, \vec{\alpha}) \in D_{M^{\leq}(P)}$$

Y diremos que $M^{\leq}(P)$ se obtiene por *minimización de variable alfabética* a partir de P .

Combo 12

Defina cuándo un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es llamado Σ -computable, cuándo es llamado Σ -enumerable y defina "el programa \mathcal{P} enumera a S ".

Resolución

• Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ -computable si $\chi_S^{\omega^n \times \Sigma^{*m}}$ es Σ -computable. Es decir, es Σ -computable si y solo si hay un programa $\mathcal{P} \in Pro^{\Sigma}$ que computa a $\chi_S^{\omega^n \times \Sigma^{*m}}$:

- Si $(\vec{x}, \vec{\alpha}) \in S$, entonces \mathcal{P} se detiene partiendo de $||x_1, \dots, x_n, \alpha_1, \dots, \alpha_m||$ y la variable N1 queda con contenido igual a 1
- Si $(\vec{x}, \vec{\alpha}) \notin S$, entonces \mathcal{P} se detiene partiendo de $||x_1, \dots, x_n, \alpha_1, \dots, \alpha_m||$ y la variable N1 queda con contenido igual a 0

Decimos que \mathcal{P} decide la pertenencia a S respecto al conjunto $\omega^n \times \Sigma^{*m}$

- Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ -enumerable si es vacío o existe una función $F: \omega \rightarrow \omega^n \times \Sigma^{*m}$ tal que $I_F = S$ y $F_{(i)}$ sea una función Σ -computable para todo $i \in 1, \dots, n+m$
- Por *propiedad*, sabemos que: Sea $S \subseteq \omega^n \times \Sigma^{*m}$ un conjunto no vacío, entonces son equivalentes:
 - S es Σ -enumerable
 - Hay un programa $\mathcal{P} \in Pro^\Sigma$ tal que
 - $\forall x \in \omega$, \mathcal{P} se detiene partiendo de $\|x\|$ y llega a un estado de la forma $((x_1, \dots, x_n, y_1, \dots), (\alpha_1, \dots, \alpha_m, \beta_1, \dots))$ con $(x_1, \dots, x_n, \alpha_1, \dots, \alpha_m) \in S$
 - $\forall (x_1, \dots, x_n, \alpha_1, \dots, \alpha_m) \in S$, $\exists x \in \omega$ tal que \mathcal{P} se detiene partiendo de $\|x\|$ y llega a un estado de la forma $((x_1, \dots, x_n, y_1, \dots), (\alpha_1, \dots, \alpha_m, \beta_1, \dots))$

Decimos que \mathcal{P} enumera a S

Combo 13

Defina:

1. $i^{n,m}$
2. $E_{\#}^{n,m}$
3. $E_{*}^{n,m}$
4. $E_{\#j}^{n,m}$
5. $E_{*j}^{n,m}$
6. $Halt^{n,m}$
7. $T^{n,m}$
8. $AutoHalt^\Sigma$
9. Los conjuntos A y N

Resolución

- Sean $n, m \in \omega$, definimos las siguientes funciones:

$$\begin{aligned} i^{n,m} &: \omega \times \omega^n \times \Sigma^{*m} \times Pro^\Sigma \rightarrow \omega \\ E_{\#}^{n,m} &: \omega \times \omega^n \times \Sigma^{*m} \times Pro^\Sigma \rightarrow \omega^{[N]} \\ E_{*}^{n,m} &: \omega \times \omega^n \times \Sigma^{*m} \times Pro^\Sigma \rightarrow \Sigma^{[N]} \end{aligned}$$

de modo 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 descripción instantánea que se obtiene luego de correr \mathcal{P} una cantidad t de pasos partiendo del estado $\|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\|$.

Si las definimos formalmente, podemos hacerlo de forma recursiva:

$$(i^{n,m}(0, \vec{x}, \vec{\alpha}, \mathcal{P}), E_{\#}^{n,m}(0, \vec{x}, \vec{\alpha}, \mathcal{P}), E_{*}^{n,m}(0, \vec{x}, \vec{\alpha}, \mathcal{P})) = (1, (x_1, \dots, x_n, 0, \dots), (\alpha_1, \dots, \alpha_n$$

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

- Definimos también las funciones

$$E_{\#j}^{n,m} : \omega \times \omega^n \times \Sigma^{*m} \times Pro^{\Sigma} \rightarrow \omega$$

$$E_{*j}^{n,m} : \omega \times \omega^n \times \Sigma^{*m} \times Pro^{\Sigma} \rightarrow \Sigma^*$$

que marcan el valor de la j -ésima componente de $E_{\#}^{n,m}$ y $E_{*}^{n,m}$, respectivamente. Es decir:

$$E_{\#j}^{n,m} = p_j^{n,m} \circ E_{\#}^{n,m}$$

$$E_{*j}^{n,m} = p_j^{n,m} \circ E_{*}^{n,m}$$

- Dados $n, m \in \omega$, definimos $Halt^{n,m} = \lambda t \vec{x} \vec{\alpha} \mathcal{P} [i^{n,m}(t, \vec{x}, \vec{\alpha}, \mathcal{P}) = n(\mathcal{P}) + 1]$
 - Básicamente, $Halt^{n,m}$ es un predicado que dice si \mathcal{P} se detiene luego de t pasos partiendo del estado $\|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\|$.
- Definimos $T^{n,m} = M(Halt^{n,m})$
 - $D_{T^{n,m}} = \{(\vec{x}, \vec{\alpha}, \mathcal{P}) : \mathcal{P} \text{ se detiene partiendo de } \|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\|\}$
 - Para $(\vec{x}, \vec{\alpha}, \mathcal{P}) \in D_{T^{n,m}}$, $T^{n,m}(\vec{x}, \vec{\alpha}, \mathcal{P})$ indica la cantidad de pasos necesarios para que \mathcal{P} se detenga partiendo de $\|x_1, \dots, x_n, \alpha_1, \dots, \alpha_m\|$.
- Cuando $\Sigma \supseteq \Sigma_p$, podemos definir $AutoHalt^{\Sigma} = \lambda \mathcal{P} [(\exists t \in \omega) Halt^{0,1}(t, \mathcal{P}, \mathcal{P})]$
 - Notar que $D_{AutoHalt^{\Sigma}} = Pro^{\Sigma}$ y que $\forall \mathcal{P} \in Pro^{\Sigma}$, $AutoHalt^{\Sigma}(\mathcal{P}) = 1$ sii \mathcal{P} se detiene partiendo del estado $\|\mathcal{P}\|$.
- Supongamos $\Sigma \supseteq \Sigma_p$. Entonces $A = \{\mathcal{P} \in Pro^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 1\}$ y $N = \{\mathcal{P} \in Pro^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 0\}$.

Combo 14

Explique en forma detallada la notación lambda.

Resolución

- Una expresión es *lambdificable* con respecto a Σ si cumple las siguientes características:
 - Involucra variables numéricas (que se valuaran en números de ω), y variables alfabéticas (que se valuaran en palabras del alfabeto previamente fijado)
 - En cuanto a notación, las numéricas son con letras latinas minúsculas (x, y, z) y las alfabéticas con letras griegas minúsculas (α, β, γ)

- Para ciertas valuaciones de sus variables la expresión puede *no* estar definida (por ejemplo, $Pred(|\alpha|)$ para $\alpha = \varepsilon$)
- Sea E la expresión, los valores que asuma cuando hayan sido asignados los valores de ω a sus variables numéricas y valores de Σ^* a sus variables alfabéticas, deberán ser *siempre* elementos de $O \in \{\omega, \Sigma^*\}$ (es decir, no puede tomar valores mixtos)
- La expresión puede involucrar lenguaje coloquial castellano (i.e., no únicamente operaciones matemáticas). Por ejemplo, "el menor número primo que es mayor que x "
- A las *expresiones booleanas* (como $x = 0$), se les considerará que asumen valores de $\{0, 1\} \subseteq \omega$
- **Definición:** sea Σ un alfabeto finito fijo, E una expresión lambdificable respecto a Σ y $x_1, \dots, x_n, \alpha_1, \dots, \alpha_m$ variables distintas tales que las numéricas que ocurren en E están en $\{x_1, \dots, x_n\}$ y las alfabéticas en $\{\alpha_1, \dots, \alpha_m\}$, entonces $\lambda x_1. \dots x_n \alpha_1. \dots \alpha_m [E]$ denota la función definida por:
 - $D_{\lambda x_1. \dots x_n \alpha_1. \dots \alpha_m [E]} = \{(k_1, \dots, k_n, \beta_1, \dots, \beta_m) \in \omega^n \times \Sigma^{*m} : E \text{ está definida cuando asignamos a cada } x_i \text{ el valor } k_i, \text{ y a cada } \alpha_i, \text{ el valor } \beta_i\}$
 - $\lambda x_1. \dots x_n \alpha_1. \dots \alpha_m [E](k_1, \dots, k_n, \beta_1, \dots, \beta_m) = \text{valor que asume o representa } E \text{ cuando asignamos a cada } x_i \text{ el valor } k_i, \text{ y a cada } \alpha_i, \text{ el valor } \beta_i$

Combo 15

Dada una función $f : D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$, describa qué tipo de objeto es y qué propiedades debe tener el macro:

$$[V2 \leftarrow f(V1, W1)]$$

Resolución

Dada la función $f : D_f \subseteq \omega \times \Sigma^* \rightarrow \omega$, el macro $[V2 \leftarrow f(V1, W1)]$ es un objeto de tipo **PALABRA**.

Para que el macro $[V2 \leftarrow f(V1, W1)]$ sea válido (i.e., exista en el lenguaje S^Σ), debe cumplir las siguientes propiedades:

- Las variables oficiales de M son $V1, V2, W1$
- M no tiene labels oficiales
- Si reemplazamos:
 - las variables oficiales de M por variables concretas $\overline{Nk_1}, \overline{Nk_2}, \overline{Pj_1}$,
 - las variables auxiliares de M por variables concretas distintas de a dos y NO pertenecientes a $\{\overline{Nk_1}, \overline{Nk_2}, \overline{Pj_1}\}$,

- los labels auxiliares de M por labels concretos distintos de a dos, entonces la palabra obtenida es un programa de S^Σ que denotaremos con $[\overline{Nk_2} \leftarrow f(\overline{Nk_1}, \overline{Pj_1})]$ y tiene la siguiente propiedad:
- Si corremos $[\overline{Nk_2} \leftarrow f(\overline{Nk_1}, \overline{Pj_1})]$ partiendo de un estado e que asigne a $\overline{Nk_1}, \overline{Pj_1}$ los valores x_1, α_1 respectivamente, entonces independientemente de los valores que les asigne e a las demás variables, se dará que:
 - Si $(x_1, \alpha_1) \notin D_f$, entonces $[\overline{Nk_2} \leftarrow f(\overline{Nk_1}, \overline{Pj_1})]$ **no** se detiene partiendo de e
 - Si $(x_1, \alpha_1) \in D_f$, entonces $[\overline{Nk_2} \leftarrow f(\overline{Nk_1}, \overline{Pj_1})]$ se detiene partiendo de e y llega a un estado e' que cumple que:
 - e' le asigna a $\overline{Nk_2}$ el valor $f(x_1, \alpha_1)$
 - e' solo puede diferir de e en los valores que le asigna a $\overline{Nk_2}$ o a las variables que fueron a reemplazar a las variables auxiliares de M

Combo 16

Dado un predicado $P : D_P \subseteq \omega^n \times \Sigma^{*m} \rightarrow \omega$, describa qué tipo de objeto es y qué propiedades debe tener el macro:

[IF $P(V1, W1)$ GOTO $A1$]

Resolución

Dado el predicado $P : D_P \subseteq \omega \times \Sigma^* \rightarrow \omega$, el macro [IF $P(V1, W1)$ GOTO $A1$] es un objeto de tipo **PALABRA**.

Para que el macro [IF $P(V1, W1)$ GOTO $A1$] sea válido (i.e., exista en el lenguaje S^Σ), debe cumplir las siguientes propiedades:

- Las variables oficiales de M son $V1, W1$
- $A1$ es el único label oficial de M
- Si reemplazamos:
 - las variables oficiales de M por variables concretas $\overline{Nk_1}, \overline{Pj_1}$,
 - el label oficial $A1$ por el label concreto \overline{Lk} ,
 - las variables auxiliares de M por variables concretas distintas de a dos y NO pertenecientes a $\{\overline{Nk_1}, \overline{Pj_1}\}$,
 - los labels auxiliares de M por labels concretos distintos de a dos y ninguno de ellos igual a \overline{Lk} ,
 entonces la palabra obtenida es un programa de S^Σ que denotaremos con $[\text{IF } P(\overline{Nk_1}, \overline{Pj_1}) \text{ GOTO } \overline{Lk}]$ y tiene la siguiente propiedad:
- Si corremos $[\text{IF } P(\overline{Nk_1}, \overline{Pj_1}) \text{ GOTO } \overline{Lk}]$ partiendo de un estado e que asigne a $\overline{Nk_1}, \overline{Pj_1}$ los valores x_1, α_1 respectivamente, entonces independientemente de los valores que les asigne e a las demás variables, se dará que:

- Si $(x_1, \alpha_1) \notin D_P$, entonces $[\text{IF } P(\overline{Nk_1}, \overline{Pj_1}) \text{ GOTO } \overline{Lk}]$ **no** se detiene partiendo de e
- Si $(x_1, \alpha_1) \in D_P$ y $P(x_1, \alpha_1) = 1$, entonces, luego de una cantidad finita de pasos, $[\text{IF } P(\overline{Nk_1}, \overline{Pj_1}) \text{ GOTO } \overline{Lk}]$ direcciona al label \overline{Lk} quedando en un estado e' que solo puede diferir de e en los valores que le asigna a las variables que fueron a reemplazar a las variables auxiliares de M
- Si $(x_1, \alpha_1) \in D_P$ y $P(x_1, \alpha_1) = 0$, entonces, luego de una cantidad finita de pasos, $[\text{IF } P(\overline{Nk_1}, \overline{Pj_1}) \text{ GOTO } \overline{Lk}]$ se detiene partiendo de e quedando en un estado e' que solo puede diferir de e en los valores que le asigna a las variables que fueron a reemplazar a las variables auxiliares de M