

INET
Lógica
Práctico Funciones Recursivas

Ejercicio 1

Sea Σ un alfabeto. La función $cant : \Sigma \times \Sigma^* \rightarrow N$ calcula recursivamente la cantidad de ocurrencias de un símbolo en una frase.

- i) $cant(x, \epsilon) = 0$ con $x \in \Sigma$
- ii) $cant(x, y\alpha) = 1 + cant(x, \alpha)$ si $x = y$ con $x, y \in \Sigma$
- iii) $cant(x, y\alpha) = cant(x, \alpha)$ si $x \neq y$ con $x, y \in \Sigma$

1. Calcule $cant(a, aba)$ y $cant(a, bc)$ aplicando paso por paso la definición de la función.
2. Defina $\Sigma \times \Sigma^*$ inductivamente de forma que $cant$ siga el esquema de recursión primitiva.

Ejercicio 2

Sea Σ el alfabeto $\{a, b, c, d\}$. Sea L_3 el lenguaje definido por las siguientes reglas:

1. $\epsilon \in L_3$
2. Si $w \in L_3$ entonces $abwcd \in L_3$

Defina recursivamente la función $duplicar : L_3 \rightarrow \Sigma^*$ que dada una palabra $w \in L_3$ calcula la palabra resultante de duplicar cada letra de w .

Por ejemplo: $duplicar(ababcdcd) = aabbaabbccddccdd$

Ejercicio 3

Sea Σ el alfabeto $\{a, b\}$. Defina recursivamente la función $copiar : N \times \Sigma^* \rightarrow \Sigma^*$ que dada una frase α de Σ^* y un número natural n arma una palabra compuesta de n copias de α .

Por ejemplo: $copiar(3, ab) = ababab$

Ejercicio 4

Sea Σ un alfabeto. Defina recursivamente la función $inversa : \Sigma^* \rightarrow \Sigma^*$ que dada una frase de Σ^* devuelve la frase dada vuelta.

Por ejemplo: $inversa(abc) = cba$

Ejercicio 5

Considere el lenguaje $L_2 \subseteq \Sigma^*$ donde $\Sigma = \{a, b, c\}$:

1. $b \in L_2$
2. Si $w \in L_2$ entonces $awc \in L_2$

Defina las siguientes funciones:

- a) $largo : L_2 \rightarrow \mathbb{N}$ que aplicada a una palabra devuelve el largo de la palabra (cantidad de símbolos).

Por ejemplo: $largo(abc) = 3$

- b) $aporc : L_2 \rightarrow \Sigma^*$ tal que $apor$ aplicada a una palabra cambia las letras a por letras c y las letras c por letras a . Las b quedan incambiadas.

Por ejemplo: $aporc(aabcc) = ccbaa$

Ejercicio 6

Sea $\Sigma = \{a, b, c, \dots, x, y, z\}$. Defina recursivamente la función $par : \Sigma^* \rightarrow \{0, 1\}$ de forma que $par(\alpha) = 1$ si y solo si α tiene un número par de símbolos.

Ejercicio 7

Defina la función $palindrome : \{a, b\}^* \rightarrow \{0, 1\}$ tal que aplicada a una palabra devuelve 1 si la palabra es capicua y 0 si no lo es.

Ejercicio 8

Sea Σ un alfabeto y la función $misterio : \Sigma^* \times N \times N \rightarrow \Sigma^*$ definida por:

1. $misterio(\epsilon, n, m) = \epsilon$
2. $misterio(\alpha, n, 0) = \epsilon$ con $\alpha \in \Sigma^*$
3. $misterio(x\alpha, 0, m+1) = xmisterio(\alpha, 0, m)$ con $x \in \Sigma, \alpha \in \Sigma^*$
4. $misterio(x\alpha, n+1, m+1) = misterio(\alpha, n, m+1)$ con $x \in \Sigma, \alpha \in \Sigma^*$

a) Calcule $misterio(alfabetos, 4, 4)$, $misterio(alfabetos, 0, 4)$, $misterio(alfabetos, 3, 6)$ y $misterio(alfabetos, 0, 8)$

b) Qué implementa la función $misterio$?