

Fundamentos del Diseño de Lenguajes de Programación - 2024
Práctico Nro. 2
Aspectos Formales de los Lenguajes de Programación

Ejercicio 1.

Dar al menos 2 ejemplos de palabras de distinta longitud, que pertenezcan a cada uno de los siguientes lenguajes expresados por comprensión:

- $L_1 = \{a^i b \mid i > 0\}$
- $L_2 = \{w \in \{a, b\}^* / w \text{ contiene cantidad impar de } b\text{'s}\}$
- $L_3 = \{(a^i)^{2*i} / i > 0\}$

Ejercicio 2.

Determinar el valor de verdad de las siguientes afirmaciones. La notación $L(ER)$ significa el lenguaje denotado por la expresión regular ER:

$$\begin{array}{ll} aaab \in L(a^*b) & ab \notin L(aba^*) \\ abab \in L(a^* + b^*) & abba \in L((ab + ba)^*) \end{array}$$

Ejercicio 3.

Decir para cada una de las siguientes ER's si denotan el lenguaje especificado:

- ¿ $ab(a + b)^*$ denota a $L_1 = \{w \in \{a, b\}^* / w \text{ comienza con } ab\}$?
- ¿ $(ab + ba)^*$ denota a $L_2 = \{w \in \{a, b\}^* / w \text{ contiene la subcadena } ab \text{ o } ba\}$?
- ¿ bb^*abb^* denota a $L_3 = \{b^n ab^m \mid n \text{ es impar, } m \geq 1\}$?

Ejercicio 4.

Para cada uno de los siguientes lenguajes definidos sobre el alfabeto $\Sigma = \{a, b\}$, construir una expresión regular que los denote:

- $L_1 = \{w \in \{a, b\}^* / w \text{ finaliza con } ab\}$
- $L_2 = \{w \in \{a, b\}^* \mid w \text{ contiene solamente dos } a\text{'s}\}$
- $L_3 = \{b^3 a^i b^j \mid 0 < i < 3, j \geq 0\}$

Ejercicio 5.

A partir de los siguientes lenguajes:

- $L_1 = \{w \in \{a, \dots, b, A, \dots, Z, 0\dots, 9\} / w \text{ es un identificador}\}$
- $L_2 = \{w \in \{0\dots, 9\} / w \text{ es un número entero sin signo}\}$

- a. Dar ER's que denoten a cada lenguaje.
- b. Reescribir las ER's del ítem anterior utilizando el metalenguaje explicado en el anexo del práctico.
- c. Utilizar la herramienta online *Regular Expressions 101*, cuya dirección Web es <https://regex101.com/>, y probar las ER's del ítem previo realizando el testeo con códigos Java. Para tal fin, seleccionar en *Flavor* Java 8.

Ejercicio 6.

A partir de la siguiente gramática:

$$G = \langle \{S, I\}, \{\{, \}, i\}, P, S \rangle$$

$$P = \{S \rightarrow \{IS\} \mid \{I\} \\ I \rightarrow i\}$$

- a. Determinar cuáles de las siguientes cadenas pueden ser generadas por la gramática, utilizando los 2 criterios de selección del no terminal a expandir:
 - $\{i\{i\}\}$
 - $\{i\{\}\}$
- b. Construir un árbol de derivación para cada una de las siguientes cadenas:
 - $\{i\}$
 - $\{i\{i\{i\}\}\}$
- c. Describir formalmente la gramática.

Ejercicio 7.

Construir GLCs que generen los siguientes lenguajes:

- a. $L_1 = \{a^i b^i \mid i > 0\}$
- b. $L_2 = \{a^i b^j c^{2*i} \mid i, j > 0\}$
- c. $L_3 = \{c^i d^s u^r f^k \mid i, k, s > 0, r = s + 1, i = 2 * k\}$

Verificar cada una de las gramáticas construidas derivando cadenas pertenecientes a cada lenguaje.

Ejercicio 8.

En el lenguaje Prolog las estructuras pueden ser anidadas, con todos los niveles que se requieran ejemplos: $a(a(1))$, $ab(ab(ab(1)))$. A los fines de limitar las cadenas pertenecientes al lenguaje, se deben tener en cuenta las siguientes consideraciones: El nombre de la estructura puede ser un nombre, con solamente las tres primeras letras minúsculas del alfabeto: a,b y c, ejemplos: $aabbc$, bc , $abcabc$ y el único elemento que puede tener la estructura más interna es el valor 1. Ejemplos: $abc(1)$ y $aabbc(1)$.

La siguiente es descripción del lenguaje L :

$$L = \{w \in \{a, b, c, (,), 1\}^* / w \text{ son estructuras de prolog de un componente con las especificaciones dadas} \}$$

Construir una gramática libre del contexto (GLC) en notación BNF que genere el lenguaje L .

Ejercicio 9.

Convertir la siguiente BNF a BNFE:

$\langle \text{Decl} \rangle ::= \langle \text{Tipo} \rangle \langle \text{Ident} \rangle \langle \text{Esp} \rangle ; \langle \text{Decl} \rangle \mid \lambda$
 $\langle \text{Tipo} \rangle ::= \text{char} \mid \text{int}$
 $\langle \text{Esp} \rangle ::= \langle \text{Constante} \rangle \mid [\langle \text{Constante} \rangle] \mid \lambda$

Ejercicio 10.

En el lenguaje C es posible que una única sentencia de asignación realice varias asignaciones de forma múltiple, a todas las variables previamente declaradas. Asuma que todas variables ya han sido declaradas de tipo *char* y los únicos nombres posibles para las variables son: n , r y t y solo las siguientes constantes carácter se pueden asignar: $'n'$, $'r'$ y $'t'$, también es posible que las constantes caracteres estén precedidas por \backslash indicando que son caracteres de escape. Dar una BNFE que genere la operación de asignación múltiple en C con las consideraciones dadas. Ejemplos de cadenas válidas:

$n='t';$
 $r=t='\backslash n';$
 $n=t=r='n';$

Ejercicio 11.

Para los siguientes AFD's provea el lenguaje que aceptan con notación de conjunto:

