Tema 1: Lenguajes Regulares

Lección 1.1

Expresiones regulares

Jorge Bernad, Elvira Mayordomo

Índice

- Una expresión aritmética es una forma de expresar números mediante operadores

$$(2+3)*(4+7)$$

- El resultado de una expresión aritmética es un número.
- Las expresiones regulares son una forma de expresar lenguajes mediante operadores

$$a^* = \{\epsilon, a, aa, aaa, \dots\}$$
$$a+b = \{a, b\}$$

- El resultado de una expresión regular es un lenguaje.

Operadores en expresiones regulares

En una expresión regular podremos utilizar los operadores:

- Unión: $(a + b)$
- Concatenación: $(a + b) \cdot (a + b)$ equivalente $(a + b)(a + b)$
- Estrella de Kleene: a^*

Ejemplos

- $(a+b)(a+b) = \{aa, ab, ba, bb\}$
- $a^*b^* = \{\epsilon, a, aa, \ldots, b, bb, \ldots, ab, aab, abb, \ldots\}$
- $a + b^* = \{a, \epsilon, b, bb, bbb, \ldots\}$

Definición formal expresión regular

Definición

Una expresión r se dice que es una **expresión regular (e.r.)** si es igual a:

- $r = a$, donde a es un símbolo para algún alfabeto Σ.
- $r = \epsilon$
- $r = \emptyset$
- $r = (r_1 + r_2)$, donde r_1 y r_2 son expresiones regulares.
- $r = (r_1 \cdot r_2)$, donde r_1 y r_2 son expresiones regulares.
- $r = (r_1^*)$, donde r_1 es una expresión regular.

Observaciones

- Al igual que en las expresiones aritméticas, no podremos todos los paréntesis.
- Supondremos una precedencia en los operadores: $*$, $\cdot$, $+$
- $((a \cdot (b^*)) + (b^*)) \equiv ab^* + b^*$
- El operador concatenación $\cdot$, lo podemos suprimir.
- También utilizaremos el operador r^+, como una abreviatura de

$$r^+ = rr^*$$

- ab^+ describe $\{w \in \Sigma^* \mid w \text{ es una } a \text{ seguida de una o más } b\}$

Lenguaje representado por una e.r.

Definición

Dada una expresión regular r, el **lenguaje representado** (o **descrito**) por r se denota por $L(r)$ y es igual a

- $L(a) = \{a\}$, si a es un símbolo de un alfabeto Σ.
- $L(\epsilon) = \{\epsilon\}$
- $L(\emptyset) = \emptyset$
- $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
- $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$
- $L(r^*) = L(r)^*$

Ejemplos

- $L(a^*) = \{a\}^* = \{a^n \mid n \in \mathbb{N}\}$
- $L((ab^*)(a^*b)) = \{a\}\{b\}^* \cdot \{a\}^*\{b\} = \{ab^n a^m b \mid n, m \in \mathbb{N}\}$
- $L((1+0)1^+) = \{1, 0\} \cdot \{1\}\{1\}^* = \{1^n \mid n \geq 2\} \cup \{01^n \mid n \geq 1\}$
- $L(\emptyset \cdot 1^*) = \emptyset \cdot \{1\}^* = \emptyset$

Expresiones regulares equivalentes

Definción

Dos expresiones regulares r_1 y r_2 son **equivalentes**, si representan el mismo lenguaje, $L(r_1) = L(r_2)$. Se denotará por $r_1 = r_2$.

- $(a + b)a^* = aa^* + ba^*$
- $(a + ba)^* = (a^*(ba)^*)^*$

Expresiones regulares Flex

Los sistemas operativos y lenguajes de programación ofrecen herramientas para analizar textos mediante expresiones regulares. Ejemplos:

- Comando *grep* en sistemas Unix
- Librería *regex* de C++
- *Flex* (se usará en prácticas)

Operación	**e.r. Teoría**	**e.r. Flex**
Concatena	$\cdot$ ó nada	nada
Unión	$+$	$\vert$
Kleene	*	$*$
$+$	$^+$	$+$
Paréntesis	( )	( )
$a, b, c, 0, 1, 2$	$(a + b + c + 0 + 1 + 2)$	$[abc012]$ ó $[a - c0 - 2]$
cadena vacía ó a	$\epsilon + a$	$a?$
Cadena vacía	ϵ	Sin equivalencia (ver $*$ ó ?)
Conjunto vacío	$\emptyset$	Sin equivalencia

Analizadores lexicográfico

- Un **analizador lexicográfico** es un programa que recorre un texto buscando fragmentos que cumplan un patrón y realiza acciones con estos fragmentos.
- Los fragmentos que se buscan se expresan con e.r.
- Por ejemplo, un programa que busca todas las apariciones de varios espacios seguidos y las cambia por uno solo.
- En prácticas usaréis Flex, un creador de analizadores lexicográficos.

Referencias

- Kelly: capítulo 2, sección 2.2
- Sipser: capítulo 1, sección 1.3