Expresiones Regulares

Alan Reyes-Figueroa

Teoría de la Computación **(Aula 03b) 10.julio.2023**

Expresiones regulares
Árboles sintácticos
Notación Polaca Reversa

Operaciones para lenguajes

Dados lenguajes L, L_1 y L_2, tenemos

- La *concatenación* L_1L_2 es el lenguaje

  $L_1L_2 = \{w_1w_2 : w_1 \in L_1 , w_2 \in L_2\}$.
- La *unión* $L_1 \cup L_2$ es el lenguaje

  $L_1 \cup L_2 = \{w : w \in L_1$ ó $w \in L_2\}$.
- La *cerradura de Kleene* de L es

  $L^* = \{w_1w_2 \dots w_n : w_i \in L, n \geq 0\}$.

Operaciones para lenguajes

- La *cerradura positiva* de L es el lenguaje

$$L^+ = \{w_1 w_2 \ldots w_n : w_i \in L, n \geq 1\}.$$

<u>Ejemplo</u>: $L_1 = \{a, aa\}$ $L_2 = \{\varepsilon, b, c\}$.

$L_1 \cup L_2 = \{\varepsilon, a, aa, b, c\}$.

$L_1 L_2 = \{a, ab, ac, aa, aab, aac\}$.

$L_2^* = \{\varepsilon, b, c, bb, bc, cc, cb, bbb, bbc, \ldots\}$.

Expresiones regulares

Una *expresión regular* es una representación de un lenguaje (**Obs:** no de cualquier lenguaje)

Los lenguajes que son representables mediante expresiones regulares se llaman *lenguajes regulares*.

Dado un alfabeto Σ, para representar un lenguaje regular usamos los símbolos en Σ, y ciertos operadores especiales:

Expresiones regulares

- ab ó a·b, para la concatenación
- a u b, ó a | b ó a+b, para la unión
- a^* para la cerradura de Kleene
- a^+ para la cerradura positiva
- (,), [,], para definir agrupaciones y jerarquías

También se usan otros símbolos como abreviaturas: $[a_1, a_2, \ldots, a_n]$

$[a_1 - a_n]$

Expresiones regulares

Las expresiones regulares en un alfabeto Σ se construyen siguiendo las

1. ε y cualquier elemento de Σ es una expresión regular.
2. Si α y β son expresiones regulares, también lo es $\alpha\beta$.
3. Si α y β son expresiones regulares, también lo es $\alpha \mid \beta$.
4. Si α es expresión regular, también lo son α^* y α^+.
5. Sólo las reglas 1-4 generan expresiones regulares.

Precedencia de operadores

1. () operadores de agrupación se aplican primero.
2. * se aplica antes que | y que .
3. . Concatenación precede a |.
4. | se aplica al final.

Ejemplos:

ab*	es equivalente a	a(b*)
ab|c	es equivalente a	(ab)|c
a|b*	es equivalente a	a|(b*)

Expresiones regulares

Ejemplo: En $\Sigma = \{0,1\}$

$\alpha = (0|1)^*0$ es expresión regular.

Representa todas las cadenas terminadas en 0.

Si L es el lenguaje representado por la expresión α, en L tenemos las cadenas

0	00	000	100	0000	…
	10	010	110	0010	

Expresiones regulares

Ejemplo: En $\Sigma = \{a,b\}$

$\beta = b^*(abb^*)(a|\varepsilon)$ es expresión regular.

Representa a las cadenas que comienzan con un número cualquiera de b's, luego tiene ab, luego tiene cualquier número de b's luego terminan en a ó en ε.

En el lenguaje L representado por β tenemos

ab	bab		baba	bbab
	abb	aba	abbb	babb

Árboles sintácticos

Cada expresión regular tiene asociado un **árbol sintáctico** o (*parse tree*), el cual representa de forma visual la estructura de la regexp

(a|b)*ab*

```
                   |
                    .
               /          \
            .                |
        /       \            |
     *             |         |
     |             |         |
     ∪             |         *
  /      \         |         |
a          b       a         b
```

Notaciones

Notación infix: La que usamos regularmente. Los operadores binarios van en medio de los términos.

(a | b)* a b b*

Notación prefix: También se llama notación polaca (PN). Los operadores preceden a los términos.

• • • * | a b b * b

Notación postfix: También se llama notación polaca revertida (RPN), o notación de Lukasiewicz. Los operadores van después.

a b | * a • b • b * •

Ejercicios

Para cada una de las expresiones regulares siguientes, construir:

- El árbol sintáctico
- La expresión en notación prefix y postfix
- El AFD que representa la expresión
- La tabla de transiciones

Ejercicios

Aquí $\Sigma = \{a,b,c\}$

1) R = a*
2) R = a|b
3) R = bb* | a
4) R = abc | bac
5) R = (a|b)*
6) R = a*|b*