



IC-5701

Compiladores e Intérpretes

Profesor:

Ing. Allan Rodríguez Dávila, MGP

Análisis Léxico

Expresiones Regulares

Autómatas Finitos

Conversión

# Expresiones Regulares

# Expresiones Regulares

- **Notación** para expresar un conjunto de strings de símbolos terminales.

Sintaxis	Descripción
' '	Alternativas separadas
'*'	Ítem previo aparece zero o muchas veces
'+'	Ítem previo aparece una o muchas veces
'.'	Un carácter cualquiera
'(' y ')'	Paréntesis de agrupamiento

# Expresiones Regulares

- Notación para especificar patrones de lexemas.

Sintaxis	Descripción
'^'	Inicio de Línea
'\$'	Fin de línea
'\'	Carácter de escape
'?'	Ítem previo aparece cero o una vez
'[' y ']'	Cualquier carácter dentro de los paréntesis
'{'n,m'}	Ítem previo aparece de n a m veces

# Expresiones Regulares

Acción	Expresión Regular	Descripción
Vacío	$\epsilon$	String vacío
Un ítem	$t$	String de $t$ sólo
Concatenación	$XY$	Concatenación de cualquier string generado por $X$ y cualquier string generado por $Y$
Alternativa	$X/Y$	Cualquier string generado ya sea por $X$ o por $Y$
Iteración	$X^*$	Concatenación de cero o más string generados por $X$
Agrupamiento	$(X)$	Cualquier string generado por $X$

# Expresiones Regulares

Expresión Regular	Conjunto Resultante
$Mr \mid Ms$	Mr, Ms
$M(r \mid s)$	Mr, Ms
$ps^*t$	pt, pst, psst, pssst, ...
$ba(na)^*$	ba, bana, banana, bananana, ...
$M(r \mid s)^*$	M, Mr, Ms, Mrr, Mrs, Mss, Mrrr, ...

# Conceptos

- Un **alfabeto** es un conjunto finito de símbolos y se representa por  $\Sigma$ .
  - $\{0,1\}$  alfabeto binario
- $|s|$  representa el **número de ocurrencias** de  $s$ .



# Conceptos

- La **cadena vacía** es presentada por  $\epsilon$ 
  - Cadena de longitud 0
  - Es la identidad en la concatenación
    - $\epsilon s = s\epsilon = s$
- Un **lenguaje** es cualquier conjunto contable de cadenas de algún alfabeto fijo.

# Conceptos

- El **prefijo** de la cadena **s** es cualquier cadena que se obtiene de **eliminar** **cero o más símbolos del final** de **s**.
- El **sufijo** de la cadena **s** es cualquier cadena que se obtiene de **eliminar** **cero o más símbolos del principio** de **s**.

# Conceptos

- Una **subcadena** de **s** se obtiene de eliminar cualquier prefijo y cualquier sufijo de **s**.
- Prefijos, sufijos y subcadenas **propios** de una cadena **s** son aquellos que no son  $\epsilon$  ni iguales a la misma cadena.

# Conceptos

- Una **subsecuencia** de **s** es cualquier cadena que se forma mediante la **eliminación de cero o más posiciones no necesariamente consecutivas** de **s**.

# Operaciones de Lenguaje

Operaciones	Definición y notación
Unión L y M	$L \cup M = \{s \mid s \text{ está en } L \text{ o en } M\}$
Concatenación L y M	$LM = \{st \mid s \text{ está en } L \text{ y } t \text{ en } M\}$
Cerradura de Kleene de L	$L^* = \bigcup_{i=0}^{\infty} L^i$
Cerradura positivo de L	$L^+ = \bigcup_{i=1}^{\infty} L^i$

# Expresiones Regulares

- Notación utilizada para describir lenguajes por medio de operadores que se aplican a los símbolos del alfabeto.
- Cada expresión regular  $r$  denota un lenguaje  $L(r)$ 
  - Ejemplo: Identificadores en C
    - `letra_(letra_|digito_)*`

# Leyes Algebraicas para REGEX

Operaciones	Definición y notación
$r   s = s   r$	$ $ es conmutativo
$r   (s   t) = (r   s)   t$	$ $ es asociativo
$r(st) = (rs)t$	La concatenación es asociativa
$r(s   t) = rs   rt; (s   t)r = sr   tr$	La concatenación se distribuye sobre $ $
$\epsilon r = r\epsilon = r$	$\epsilon$ es la identidad para la concatenación
$r^* = (r   \epsilon)^*$	$\epsilon$ se garantiza en una cerradura
$r^{**} = r^*$	$*$ es idempotente

## Repaso Portafolio #1

- Escriba las expresiones regulares para los siguientes lenguajes:
  - {a, ab, abb, abbb, abbbb, ..., ac, abc, abbc, abbbc, abbbbc, ...}
  - {olu, omu, onu, ulo, umo, uno}
  - Expresión regular de números pares (entero)
  - REGEX para los números decimales



## Portafolio # 1

- Escriba las expresiones regulares para los siguientes lenguajes:
  - Operaciones aritméticas de suma, resta, multiplicación y división con literales y/o id
  - Char con cualquier carácter
  - String con cualquier carácter
  - Asignación de una operación aritmética

# Definiciones Regulares

- Esta notación se utiliza para ponerle nombre a las expresiones regulares y utilizar en expresiones subsiguientes.
- Producciones BNF para expresiones
- Conocido también como EBNF

$$d_1 \rightarrow r_1$$

$$d_2 \rightarrow r_2$$

...

$$d_n \rightarrow r_n$$

## Definiciones Regulares

$$d_1 \rightarrow r_1$$

$$d_2 \rightarrow r_2$$

...

$$d_n \rightarrow r_n$$

- Cada  $d_i$  es un nuevo símbolo que no está en el en el alfabeto  $\Sigma$ .
- Cada  $r_i$  es una expresión regular sobre el alfabeto  $\Sigma \cup \{d_1, d_2, d_3, d_{i-1}\}$

# Definiciones Regulares

- Definición regular para identificadores

$letra\_ \rightarrow [A-Za-z\_]$

$digito\_ \rightarrow [0-9]$

$id \rightarrow letra\_ (letra\_ | digito\_)^*$

# Definiciones Regulares

- Definición regular para números sin signo

*digito*  $\rightarrow [0-9]$

*digitos*  $\rightarrow \text{digito}^+$

*fraccionOpcional*  $\rightarrow . \text{digitos} \mid \varepsilon$

*exponenteOpcional*  $\rightarrow (E(+|-|\varepsilon)\text{digitos}) \mid \varepsilon$

*numero*  $\rightarrow$

*digitos fraccionOpcional exponenteOpcional*

## Ejemplos

- Escriba las definiciones regulares para los siguientes lenguajes:
  - {int, bool, float, char, string}
  - {+, -, \*, /, %}
  - Un identificador
  - Un número entero

# Portafolio

- Cree las definiciones regulares para un lenguaje que permite:
  - **Creación de variable con asignación de literal entero o flotante**
  - **Asignación de variable con asignación de expresión aritmética binaria (suma, resta, multi, div) de literales numéricos o id**
  - **Creación de único main con única creación-asignación**

## Portafolio # 1

- Escriba la definición regular para el siguiente lenguaje:
  - Para  $\Sigma = \{a, b\}$  donde las cadenas no tienen la subcadena  $aa$  ni  $bb$
  - Todas las cadenas de letras minúsculas que contengan las cinco vocales en descendente



## Bonus

- Los lenguajes denotados por las siguientes expresiones regulares:
  - $xx(x \mid y)^*xx$
  - $n^*mn+mn^*mn+$
  - $(11^+0)+(00^*1)+$

## Bonus

- Escriba las definiciones regulares - lenguajes (EBNF) para los siguientes lenguajes:
  - Una o muchas funciones con única creación-asignación
    - Evitar el uso de operadores  $|$  y  $*$

- Programming Language Processors in Java: compilers and interpreters. Watt, David, Brown, Deryck. Pearson Education. 2000
- Compilers: principles, techniques and tools (2da. ed.). Aho, Alfred. Pearson Education. 2007

TEC | Tecnológico  
de Costa Rica