

# Expresiones Regulares

## Implementación de Métodos Computacionales (TC2037)

M.C. Xavier Sánchez Díaz  
sax@tec.mx



# Tabla de contenidos

1 Conceptos básicos de REs

2 Ejemplos de REs

3 PCRE

# ¿Qué es una expresión regular?

## Conceptos básicos de Expresiones Regulares

Recordemos que un **lenguaje** es un conjunto de palabras aceptadas por un autómata.

Si existe autómata que pueda representar al lenguaje, entonces es un lenguaje **regular**.

Como es un conjunto, podemos **describirlo** usando expresiones.

# ¿Qué es una expresión regular?

## Conceptos básicos de Expresiones Regulares

Recordemos que un **lenguaje** es un conjunto de palabras aceptadas por un autómata.

Si existe autómata que pueda representar al lenguaje, entonces es un lenguaje **regular**.

Como es un conjunto, podemos **describirlo** usando expresiones.

# ¿Qué es una expresión regular?

## Conceptos básicos de Expresiones Regulares

Recordemos que un **lenguaje** es un **conjunto** de palabras aceptadas por un autómata.

Si existe autómata que pueda representar al lenguaje, entonces es un lenguaje **regular**.

Como es un conjunto, podemos **describirlo** usando expresiones.

# ¿Qué es una expresión regular?

## Conceptos básicos de Expresiones Regulares

Recordemos que un **lenguaje** es un **conjunto** de palabras aceptadas por un autómata.

Si existe autómata que pueda representar al lenguaje, entonces es un lenguaje **regular**.

Como es un conjunto, podemos **describirlo** usando expresiones.

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

Un lenguaje  $L$  es **regular** si y sólo si se cumple al menos una de las condiciones siguientes:

- $L$  es finito.
- $L$  es la unión o concatenación de otros lenguajes regulares  $R_1$  y  $R_2$ :  
 $L = R_1 \cup R_2$  o  $L = R_1 R_2$ .
- $L$  es la cerradura de Kleene de algún lenguaje regular:  $L = R^*$ .

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

Un lenguaje  $L$  es **regular** si y sólo si se cumple al menos una de las condiciones siguientes:

- $L$  es finito.
- $L$  es la unión o concatenación de otros lenguajes regulares  $R_1$  y  $R_2$ :  
 $L = R_1 \cup R_2$  o  $L = R_1 R_2$ .
- $L$  es la cerradura de Kleene de algún lenguaje regular:  $L = R^*$ .



# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

Un lenguaje  $L$  es **regular** si y sólo si se cumple al menos una de las condiciones siguientes:

- $L$  es finito.
- $L$  es la unión o concatenación de otros lenguajes regulares  $R_1$  y  $R_2$ :  
 $L = R_1 \cup R_2$  o  $L = R_1 R_2$ .
- $L$  es la cerradura de Kleene de algún lenguaje regular:  $L = R^*$ .

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

Un lenguaje  $L$  es **regular** si y sólo si se cumple al menos una de las condiciones siguientes:

- $L$  es finito.
- $L$  es la unión o concatenación de otros lenguajes regulares  $R_1$  y  $R_2$ :  
 $L = R_1 \cup R_2$  o  $L = R_1 R_2$ .
- $L$  es la cerradura de Kleene de algún lenguaje regular:  $L = R^*$ .

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje en  $\{0,1\}$  de las palabras que empiecen con 00 pero que no tengan 11.*

00 ALGO\_NO(11)

Hay un patrón *regular* que podemos modelar usando un AF.

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje en  $\{0,1\}$  de las palabras que empiecen con 00 pero que no tengan 11.*

00 ALGO\_NO(11)

Hay un patrón *regular* que podemos modelar usando un AF.

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje en  $\{0,1\}$  de las palabras que empiecen con 00 pero que no tengan 11.*

00 ALGO\_NO(11)

Hay un patrón *regular* que podemos modelar usando un AF.

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje  $L$  de palabras formadas por  $a$  y  $b$ , pero que empiezan con  $a$ :  
 $aab, ab, a, abaa, \dots$*

- $\Sigma = \{a, b\}$
- $L = R_1 R_2$ :
  - ▶  $R_1$  = el lenguaje que contiene una  $a$ .
  - ▶  $R_2$  = el lenguaje que contiene cadenas cualesquiera de  $a$  y  $b$ .

$$L = \{a\}\{a, b\}^*$$

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje  $L$  de palabras formadas por  $a$  y  $b$ , pero que empiezan con  $a$ :  
 $aab, ab, a, abaa, \dots$*

- $\Sigma = \{a, b\}$
- $L = R_1 R_2$ :
  - ▶  $R_1$  = el lenguaje que contiene una  $a$ .
  - ▶  $R_2$  = el lenguaje que contiene cadenas cualesquiera de  $a$  y  $b$ .

$$L = \{a\}\{a, b\}^*$$

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje  $L$  de palabras formadas por  $a$  y  $b$ , pero que empiezan con  $a$ :  
 $aab, ab, a, abaa, \dots$*

- $\Sigma = \{a, b\}$

- $L = R_1 R_2$ :

- ▶  $R_1$  = el lenguaje que contiene una  $a$ .
- ▶  $R_2$  = el lenguaje que contiene cadenas cualesquiera de  $a$  y  $b$ .

$$L = \{a\}\{a, b\}^*$$



# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje  $L$  de palabras formadas por  $a$  y  $b$ , pero que empiezan con  $a$ :  
 $aab, ab, a, abaa, \dots$*

- $\Sigma = \{a, b\}$
- $L = R_1 R_2$ :
  - ▶  $R_1 =$  el lenguaje que contiene una  $a$ .
  - ▶  $R_2 =$  el lenguaje que contiene cadenas cualesquiera de  $a$  y  $b$ .

$$L = \{a\}\{a, b\}^*$$

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje  $L$  de palabras formadas por  $a$  y  $b$ , pero que empiezan con  $a$ :  
 $aab, ab, a, abaa, \dots$*

- $\Sigma = \{a, b\}$
- $L = R_1 R_2$ :
  - ▶  $R_1$  = el lenguaje que contiene una  $a$ .
  - ▶  $R_2$  = el lenguaje que contiene cadenas cualesquiera de  $a$  y  $b$ .

$$L = \{a\}\{a, b\}^*$$

# Lenguajes regulares

## Conceptos básicos de Expresiones Regulares

*El lenguaje  $L$  de palabras formadas por  $a$  y  $b$ , pero que empiezan con  $a$ :  
 $aab, ab, a, abaa, \dots$*

- $\Sigma = \{a, b\}$
- $L = R_1 R_2$ :
  - ▶  $R_1$  = el lenguaje que contiene una  $a$ .
  - ▶  $R_2$  = el lenguaje que contiene cadenas cualesquiera de  $a$  y  $b$ .

$$L = \{a\}\{a, b\}^*$$

# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.

# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.

# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.

# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.

# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.



# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.

# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.

# Expresiones Regulares

## Conceptos básicos de Expresiones Regulares

Una **expresión regular** es una representación textual de un **lenguaje regular**.

### Definición de Expresiones Regulares

Sea  $\Sigma$  un alfabeto no vacío.

- $\wedge$  es una expresión regular.
- $\emptyset$  es una expresión regular.
- Para cada  $\sigma \in \Sigma$ ,  $\sigma$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 \cup R_2$  es una expresión regular.
- Si  $R_1$  y  $R_2$  son expresiones regulares, entonces  $R_1 R_2$  es una expresión regular.
- Si  $R$  es una expresión regular, entonces  $R^*$  es una expresión regular.

# Operaciones

## Conceptos básicos de Expresiones Regulares

Las operaciones disponibles son las mismas que usábamos en autómatas, pero algunas cambian de notación:

- La RE  $\emptyset$  representa al **lenguaje vacío**  $\{\}$ .
- La **palabra vacía**  $\varepsilon$  la representamos con la expresión regular  $\hat{\phantom{x}}$ .
- La **unión** se representa usualmente con la RE  $+$ .
- La **cerradura o estrella de Kleene** de un lenguaje regular  $R$  sigue representándose con  $R^*$

# Operaciones

## Conceptos básicos de Expresiones Regulares

Las operaciones disponibles son las mismas que usábamos en autómatas, pero algunas cambian de notación:

- La RE  $\emptyset$  representa al **lenguaje vacío**  $\{\}$ .
- La **palabra vacía**  $\epsilon$  la representamos con la expresión regular  $\hat{\epsilon}$ .
- La **unión** se representa usualmente con la RE  $+$ .
- La **cerradura o estrella de Kleene** de un lenguaje regular  $R$  sigue representándose con  $R^*$

# Operaciones

## Conceptos básicos de Expresiones Regulares

Las operaciones disponibles son las mismas que usábamos en autómatas, pero algunas cambian de notación:

- La RE  $\emptyset$  representa al **lenguaje vacío**  $\{\}$ .
- La **palabra vacía**  $\varepsilon$  la representamos con la expresión regular  $\hat{\phantom{x}}$ .
- La **unión** se representa usualmente con la RE  $+$ .
- La **cerradura o estrella de Kleene** de un lenguaje regular  $R$  sigue representándose con  $R^*$

# Operaciones

## Conceptos básicos de Expresiones Regulares

Las operaciones disponibles son las mismas que usábamos en autómatas, pero algunas cambian de notación:

- La RE  $\emptyset$  representa al **lenguaje vacío**  $\{\}$ .
- La **palabra vacía**  $\varepsilon$  la representamos con la expresión regular  $\hat{\phantom{x}}$ .
- La **unión** se representa usualmente con la RE  $+$ .
- La **cerradura o estrella de Kleene** de un lenguaje regular  $R$  sigue representándose con  $R^*$

# Operaciones

## Conceptos básicos de Expresiones Regulares

Las operaciones disponibles son las mismas que usábamos en autómatas, pero algunas cambian de notación:

- La RE  $\emptyset$  representa al **lenguaje vacío**  $\{\}$ .
- La **palabra vacía**  $\varepsilon$  la representamos con la expresión regular  $\hat{\phantom{x}}$ .
- La **unión** se representa usualmente con la RE  $+$ .
- La **cerradura o estrella de Kleene** de un lenguaje regular  $R$  sigue representándose con  $R^*$



# REs sintácticamente bien formadas

## Ejemplos de REs

A nivel *implementación*, una RE es una secuencia de REs que nos dicen **cómo** debería ser el lenguaje descrito por la RE completa:

# REs sintácticamente bien formadas

## Ejemplos de REs

A nivel *implementación*, una RE es una secuencia de REs que nos dicen **cómo** debería ser el lenguaje descrito por la RE completa:

0 es una RE válida que describe al lenguaje  $\{0\}$ .

# REs sintácticamente bien formadas

## Ejemplos de REs

A nivel *implementación*, una RE es una secuencia de REs que nos dicen **cómo** debería ser el lenguaje descrito por la RE completa:

$01^*$  es una RE válida: *La **concatenación** del lenguaje  $\{0\}$  y  $\{1\}^*$ —un 0 seguido de cero o más 1s.*

# REs sintácticamente bien formadas

## Ejemplos de REs

A nivel *implementación*, una RE es una secuencia de REs que nos dicen **cómo** debería ser el lenguaje descrito por la RE completa:

$(0 + 1)$  es una RE válida: La **unión** del lenguaje  $\{0\}$  y  $\{1\}$ —o un 0 o un 1.

# REs sintácticamente bien formadas

## Ejemplos de REs

A nivel *implementación*, una RE es una secuencia de REs que nos dicen **cómo** debería ser el lenguaje descrito por la RE completa:

$(0 + 1)^*11$  es una RE válida: *o 0 o 1 cero o más veces, seguida de dos 1s.*

# REs más complejas

## Ejemplos de REs

*Diseñar una RE para el lenguaje de las palabras en  $\{0, 1\}$  que empiezan y terminan en 01:*

- 1 Estructuramos con patrones: 01 *algo* 01
- 2 Definimos restricciones: *algo* no tiene restricción alguna, puede ser 0 o 1, cero o más veces.
- 3 Generamos una RE para el patrón *algo*:  $(0 + 1)^*$
- 4 Finalmente, concatenamos las REs para hacer una nueva RE.

$$01(0 + 1)^*01$$

# REs más complejas

## Ejemplos de REs

*Diseñar una RE para el lenguaje de las palabras en  $\{0, 1\}$  que empiezan y terminan en 01:*

- 1 Estructuramos con patrones: 01 *algo* 01
- 2 Definimos restricciones: *algo* no tiene restricción alguna, puede ser 0 o 1, cero o más veces.
- 3 Generamos una RE para el patrón *algo*:  $(0 + 1)^*$
- 4 Finalmente, concatenamos las REs para hacer una nueva RE.

$$01(0 + 1)^*01$$

# REs más complejas

## Ejemplos de REs

*Diseñar una RE para el lenguaje de las palabras en  $\{0, 1\}$  que empiezan y terminan en 01:*

- 1 Estructuramos con patrones: 01 *algo* 01
- 2 Definimos restricciones: *algo* no tiene restricción alguna, puede ser 0 o 1, cero o más veces.
- 3 Generamos una RE para el patrón *algo*:  $(0 + 1)^*$
- 4 Finalmente, concatenamos las REs para hacer una nueva RE.

$$01(0 + 1)^*01$$



# REs más complejas

## Ejemplos de REs

*Diseñar una RE para el lenguaje de las palabras en  $\{0, 1\}$  que empiezan y terminan en 01:*

- 1 Estructuramos con patrones: 01 *algo* 01
- 2 Definimos restricciones: *algo* no tiene restricción alguna, puede ser 0 o 1, cero o más veces.
- 3 Generamos una RE para el patrón *algo*:  $(0 + 1)^*$
- 4 Finalmente, concatenamos las REs para hacer una nueva RE.

$$01(0 + 1)^*01$$

# REs más complejas

## Ejemplos de REs

*Diseñar una RE para el lenguaje de las palabras en  $\{0, 1\}$  que empiezan y terminan en 01:*

- 1 Estructuramos con patrones:  $01 \text{ algo } 01$
- 2 Definimos restricciones: *algo* no tiene restricción alguna, puede ser 0 o 1, cero o más veces.
- 3 Generamos una RE para el patrón *algo*:  $(0 + 1)^*$
- 4 Finalmente, concatenamos las REs para hacer una nueva RE.

$$01(0 + 1)^*01$$

# Perl-Compatible Regular Expressions

## PCRE

PCRE es un **estándar** de **expresiones regulares** muy utilizado en la industria. En PCRE existen algunos operadores que cambian su significado, así como también hay operaciones adicionales que vale la pena conocer:

- El *OR* en PCRE es `|` en lugar del `+`
- El fin de una string se marca con `$`
- Existen operaciones de *match any* y *skip any*
- Categorías de caracteres como `A-Z` o `0-9`, *whitespace* o *digit*
- Operaciones de *repeat more than x times* o *between x,y times*
- Operaciones de *positive* y *negative lookahead* y *lookbehind*

Puedes buscar más sobre ello aquí: <https://regexr.com/>