Lenguajes Regulares y Expresiones Regulares

Introducción a la Logica y la Computación (3era Parte)

Docentes: Badano, Bustos, Costamagna, Tellechea, Zigaran

Año 2024

Lenguajes Regulares

- Los **lenguajes regulares** son los lenguajes más simples dentro de la Jeraquia de Chomsky.

Lenguajes Regulares

- Los **lenguajes regulares** son los lenguajes más simples dentro de la Jeraquia de Chomsky.
- Los **lenguajes regulares** son aquellos que se pueden construir (recursivamente) a partir de ciertos **lenguajes muy básicos o atómicos**, aplicando una **cantidad finita** de ciertos operadores de lenguajes.

Lenguajes Regulares

- Los **lenguajes regulares** son los lenguajes más simples dentro de la Jeraquia de Chomsky.
- Los **lenguajes regulares** son aquellos que se pueden construir (recursivamente) a partir de ciertos **lenguajes muy básicos o atómicos**, aplicando una **cantidad finita** de ciertos operadores de lenguajes.
- Más concretamente, dichas operadores son tres: **unión, concatenación y clausura**. Es por esto, que estos operadores son conocidos como los "operadores regulares", ya que, son los que permiten construir a todos los lenguajes regulares.

Definición Recursiva de Lenguajes Regulares

Sea Σ un alfabeto, entonces:

Definición Recursiva de Lenguajes Regulares

Sea Σ un alfabeto, entonces:

Casos Base:

- $\emptyset$ es un lenguaje regular
- $\{\epsilon\}$ es un lenguaje regular
- $\{a\}$ es un lenguaje regular, con $a \in \Sigma$

Definición Recursiva de Lenguajes Regulares

Sea Σ un alfabeto, entonces:

Casos Base:

- $\emptyset$ es un lenguaje regular
- $\{\epsilon\}$ es un lenguaje regular
- $\{a\}$ es un lenguaje regular, con $a \in \Sigma$

Casos Recursivos:

Si L_1 y $L2$ son lenguajes regulares, entonces:

- $L_1 \cup L_2$ es un lenguaje regular
- $L_1 L_2$ es un lenguaje regular
- $L_1{}^*$ es un lenguaje regular

Definición Recursiva de Lenguajes Regulares

Más formalmente, podemos definir como LR_k^Σ al conjunto de los lenguajes regulares que se obtienen a partir de los casos base aplicando hasta k operadores regulares:

Definición Recursiva de Lenguajes Regulares

Más formalmente, podemos definir como LR_k^Σ al conjunto de los lenguajes regulares que se obtienen a partir de los casos base aplicando hasta k operadores regulares:

$$LR_0^\Sigma = \{\emptyset\} \cup \{\{\epsilon\}\} \cup \{\{a\} : a \in \Sigma\}$$

$$\begin{aligned} LR_{k+1}^\Sigma = LR_k^\Sigma & \cup \{L_1 \cup L_2 : L_1, L_2 \in LR_k^\Sigma\} \\ & \cup \{L_1 L_2 : L_1, L_2 \in LR_k^\Sigma\} \\ & \cup \{L^* : L \in LR_k^\Sigma\} \end{aligned}$$

Definición Recursiva de Lenguajes Regulares

Más formalmente, podemos definir como LR_k^Σ al conjunto de los lenguajes regulares que se obtienen a partir de los casos base aplicando hasta k operadores regulares:

$$LR_0^\Sigma = \{\emptyset\} \cup \{\{\epsilon\}\} \cup \{\{a\} : a \in \Sigma\}$$

$$\begin{aligned} LR_{k+1}^\Sigma = LR_k^\Sigma & \cup \{L_1 \cup L_2 : L_1, L_2 \in LR_k^\Sigma\} \\ & \cup \{L_1 L_2 : L_1, L_2 \in LR_k^\Sigma\} \\ & \cup \{L^* : L \in LR_k^\Sigma\} \end{aligned}$$

Para luego, definir como LR^Σ al conjunto de todos los lenguajes regulares:

$$LR^\Sigma = \bigcup_{k=0}^{\infty} LR_k^\Sigma$$

Ejemplos de Lenguajes Regulares

Sea $\Sigma = \{a, b\}$.

$L_1 = \{b^n aab^m : n, m \in \mathbb{N}\} \in LR^{\Sigma}$, pues

$$\begin{aligned} L_1 &= \{b^n aab^m : n, m \in \mathbb{N}\} \\ &= \{b^n : n \in \mathbb{N}\}\{aa\}\{b^n : m \in \mathbb{N}\} \\ &= \{b\}^*\{a\}\{a\}\{b\}^* \end{aligned}$$

Ejemplos de Lenguajes Regulares

Sea $\Sigma = \{a, b\}$.

$L_1 = \{b^n aab^m : n, m \in \mathbb{N}\} \in LR^{\Sigma}$, pues

$$
\begin{aligned}
L_1 &= \{b^n aab^m : n, m \in \mathbb{N}\} \\
&= \{b^n : n \in \mathbb{N}\}\{aa\}\{b^n : m \in \mathbb{N}\} \\
&= \{b\}^*\{a\}\{a\}\{b\}^*
\end{aligned}
$$

$L_2 = \{b\alpha : \alpha \in \Sigma^*\} \in LR^{\Sigma}$, pues

$$
\begin{aligned}
L_2 &= \{b\}\{\alpha : \alpha \in \Sigma^*\} \\
&= \{b\}\Sigma^* \\
&= \{b\}\{a, b\}^* \\
&= \{b\}(\{a\} \cup \{b\})^*
\end{aligned}
$$

Ejemplos de Lenguajes Regulares

Sea $\Sigma = \{a, b\}$.

$L_1 = \{b^n aab^m : n, m \in \mathbb{N}\} \in LR^\Sigma$, pues

$$\begin{aligned} L_1 &= \{b^n aab^m : n, m \in \mathbb{N}\} \\ &= \{b^n : n \in \mathbb{N}\}\{aa\}\{b^n : m \in \mathbb{N}\} \\ &= \{b\}^*\{a\}\{a\}\{b\}^* \end{aligned}$$

$L_2 = \{b\alpha : \alpha \in \Sigma^*\} \in LR^\Sigma$, pues

$$\begin{aligned} L_2 &= \{b\}\{\alpha : \alpha \in \Sigma^*\} \\ &= \{b\}\Sigma^* \\ &= \{b\}\{a, b\}^* \\ &= \{b\}(\{a\} \cup \{b\})^* \end{aligned}$$

$L_3 = \{a^n b^m : n, m \in \mathbb{N}\} \in LR^\Sigma$, pues

$$\begin{aligned} L_3 &= \{a^n : n \in \mathbb{N}\}\{b^m : m \in \mathbb{N}\} \\ &= \{a\}^*\{b\}^* \end{aligned}$$

Finitud y Regularidad

Todo lenguaje finito es un lenguaje regular.

Finitud y Regularidad

Todo lenguaje finito es un lenguaje regular.

Teorema

Si L es finito, entonces $L \in LR^{\Sigma}$.

Finitud y Regularidad

Todo lenguaje finito es un lenguaje regular.

Teorema

Si L es finito, entonces $L \in LR^{\Sigma}$.

Demo. Si L finito entonces $L = \{\alpha_1, \ldots, \alpha_k\}$ con $\alpha_i \in \Sigma^*$ y $k \geq 0$.
Si $k = 0$, entonces $L = \emptyset \therefore L$ regular por caso base.
Si $k \geq 1$, entonces $L = \{\alpha_1\} \cup \cdots \cup \{\alpha_k\} \therefore L$ regular si $\{\alpha_i\}$ es regular.

Veamos que $\{\alpha_i\} \in LR^{\Sigma}$.

Si $\alpha_i = \epsilon \therefore \{\epsilon\} \in LR^{\Sigma}$ por caso base.
Si $\alpha_i = a_1 \ldots a_r$ con $r \geq 1$, entonces
$\{\alpha_i\} = \{a_1\} \ldots \{a_r\}$, luego
$\{a_j\} \in LR^{\Sigma}$ por caso base
$\therefore \{\alpha_i\} \in LR^{\Sigma}$.

Propiedades de los Lenguajes Regulares

Los lenguajes regulares son cerrados para los operadores de conjunto y la reversa.

Propiedades de los Lenguajes Regulares

Los lenguajes regulares son cerrados para los operadores de conjunto y la reversa.

Teorema

Si $L_1, L_2 \in LR^{\Sigma}$, *entonces:*

1. $L_1 \cup L_2 \in LR^{\Sigma}$ *(trivial por definición de lenguaje regular)*
2. $L_1 \cap L_2 \in LR^{\Sigma}$
3. $L_1 - L_2 \in LR^{\Sigma}$
4. $\overline{L_1} \in LR^{\Sigma}$
5. ${L_1}^R \in LR^{\Sigma}$

Propiedades de los Lenguajes Regulares

Los lenguajes regulares son cerrados para los operadores de conjunto y la reversa.

Teorema

Si $L_1, L_2 \in LR^\Sigma$*, entonces:*

1. $L_1 \cup L_2 \in LR^\Sigma$ *(trivial por definición de lenguaje regular)*
2. $L_1 \cap L_2 \in LR^\Sigma$
3. $L_1 - L_2 \in LR^\Sigma$
4. $\overline{L_1} \in LR^\Sigma$
5. ${L_1}^R \in LR^\Sigma$

Demo. Las demostraciones las haremos más tarde en el práctico, ya que, para probarlo de forma relativamente sencilla, necesitamos un par de resultados teóricos aún no vistos hasta aquí.

Expresiones Regulares

- Las **expresiones regulares** son expresiones o fórmulas que sirven para describir lenguajes regulares (de allí su nombre) en forma muy legible y compacta.

Expresiones Regulares

- Las **expresiones regulares** son expresiones o fórmulas que sirven para describir lenguajes regulares (de allí su nombre) en forma muy legible y compacta.
- Con **expresiones regulares** podremos realizar aritmética de lenguajes regulares más fácilmente, ya que, en lugar de describir un conjunto de cadenas (regular), directamente podemos escribir la expresión regular que lo denota.

Expresiones Regulares

- Las **expresiones regulares** son expresiones o fórmulas que sirven para describir lenguajes regulares (de allí su nombre) en forma muy legible y compacta.
- Con **expresiones regulares** podremos realizar aritmética de lenguajes regulares más fácilmente, ya que, en lugar de describir un conjunto de cadenas (regular), directamente podemos escribir la expresión regular que lo denota.
- Las **expresiones regulares** también se construyen recursivamente a partir de expresiones regulares básicas o atómicas y aplicando una cierta cantidad finita de operadores.

Definición Recursiva de Expresiones Regulares

Sea Σ un alfabeto, entonces:

Definición Recursiva de Expresiones Regulares

Sea Σ un alfabeto, entonces:

Casos Base:

- $\emptyset$ es una expresión regular
- ϵ es una expresión regular
- a es una expresión regular, con $a \in \Sigma$

Definición Recursiva de Expresiones Regulares

Sea Σ un alfabeto, entonces:

Casos Base:

- $\emptyset$ es una expresión regular
- ϵ es una expresión regular
- a es una expresión regular, con $a \in \Sigma$

Casos Recursivos:
Si e_1 y e_2 son expresiones regulares, entonces:

- $e_1 + e_2$ es una expresión regular
- $e_1 e_2$ es un expresión regular
- ${e_1}^*$ es un expresión regular

Definición Recursiva de las Expresiones Regulares

Más formalmente, podemos definir como ER_k^{Σ} al conjunto de las expresiones regulares que se obtienen a partir de los casos base aplicando hasta k operadores:

$$ER_0^{\Sigma} = \Sigma \cup \{\emptyset, \epsilon\}$$

$$\begin{aligned} ER_{k+1}^{\Sigma} = ER_k^{\Sigma} & \cup \{e_1 + e_2 : e_1, e_2 \in ER_k^{\Sigma}\} \\ & \cup \{e_1 e_2 : e_1, e_2 \in ER_k^{\Sigma}\} \\ & \cup \{e^* : e \in ER_k^{\Sigma}\} \end{aligned}$$

Definición Recursiva de las Expresiones Regulares

Más formalmente, podemos definir como ER_k^{Σ} al conjunto de las expresiones regulares que se obtienen a partir de los casos base aplicando hasta k operadores:

$$ER_0^{\Sigma} = \Sigma \cup \{\emptyset, \epsilon\}$$

$$\begin{aligned} ER_{k+1}^{\Sigma} = ER_k^{\Sigma} & \cup \{e_1 + e_2 : e_1, e_2 \in ER_k^{\Sigma}\} \\ & \cup \{e_1 e_2 : e_1, e_2 \in ER_k^{\Sigma}\} \\ & \cup \{e^* : e \in ER_k^{\Sigma}\} \end{aligned}$$

Para luego, definir como ER^{Σ} al conjunto de todas las expresiones regulares:

$$ER^{\Sigma} = \bigcup_{k=0}^{\infty} ER_k^{\Sigma}$$

Lenguaje denotado por una Expresión Regular

Sea e una expresión regular, definimos como $L(e)$ al "lenguaje denotado por e" de la siguiente manera:

Lenguaje denotado por una Expresión Regular

Sea e una expresión regular, definimos como $L(e)$ al "lenguaje denotado por e" de la siguiente manera:

Casos Base:

- $L(\emptyset) = \emptyset$
- $L(\epsilon) = \{\epsilon\}$
- $L(a) = \{a\}$ con $a \in \Sigma$

Lenguaje denotado por una Expresión Regular

Sea e una expresión regular, definimos como $L(e)$ al "lenguaje denotado por e" de la siguiente manera:

<u>Casos Base:</u>

- $L(\emptyset) = \emptyset$
- $L(\epsilon) = \{\epsilon\}$
- $L(a) = \{a\}$ con $a \in \Sigma$

<u>Casos Recursivos:</u>

Si e_1 y e_2 son expresiones regulares, entonces:

- $L(e_1 + e_2) = L(e_1) \cup L(e_2)$
- $L(e_1 e_2) = L(e_1)L(e_2)$
- $L(e^*) = (L(e))^*$

Ejemplos de Expresión Regular y su Lenguaje Denotado

La expresión $e_1 = (a + b)^* a(a + b)^*$ es una expresión regular del alfabeto $\Sigma = \{a, b\}$.

Ejemplos de Expresión Regular y su Lenguaje Denotado

La expresión $e_1 = (a+b)^*a(a+b)^*$ es una expresión regular del alfabeto $\Sigma = \{a, b\}$.

Esta expresión denota al *"lenguaje de todas las cadenas que contienen al menos un símbolo a"*.

Ejemplos de Expresión Regular y su Lenguaje Denotado

La expresión $e_1 = (a + b)^* a(a + b)^*$ es una expresión regular del alfabeto $\Sigma = \{a, b\}$.

Esta expresión denota al *"lenguaje de todas las cadenas que contienen al menos un símbolo a"*.

Verifiquemos esto último aplicando la definición recursiva del lenguaje denotado por una expresión regular:

Ejemplos de Expresión Regular y su Lenguaje Denotado

La expresión $e_1 = (a+b)^* a (a+b)^*$ es una expresión regular del alfabeto $\Sigma = \{a, b\}$.

Esta expresión denota al *"lenguaje de todas las cadenas que contienen al menos un símbolo a"*.

Verifiquemos esto último aplicando la definición recursiva del lenguaje denotado por una expresión regular:

$$\begin{aligned}
L((a+b)^* a (a+b)^*) &= L((a+b)^*) L(a) L((a+b)^*)) \\
&= (L(a+b))^* L(a) (L(a+b))^* \\
&= (L(a) \cup L(b))^* L(a) (L(a) \cup L(b))^* \\
&= (\{a\} \cup \{b\})^* \{a\} (\{a\} \cup \{b\}))^* \\
&= \{a, b\}^* \{a\} \{a, b\}^* \\
&= \{\alpha_1 a \alpha_2 : \alpha_1, \alpha_2 \in \Sigma^*\} \\
&= \{\alpha \in \Sigma^* : |\alpha|_a \geq 1\}
\end{aligned}$$

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Un lenguaje es regular si y solo si es denotado por una expresión regular.

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Un lenguaje es regular si y solo si es denotado por una expresión regular.

Este resultado implica una ida y una vuelta:

1. Ida: Todo lenguaje regular tiene una expresión regular que lo denota.
2. Vuelta: Toda expresión regular denota un lenguaje regular.

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Un lenguaje es regular si y solo si es denotado por una expresión regular.

Este resultado implica una ida y una vuelta:

1. Ida: Todo lenguaje regular tiene una expresión regular que lo denota.
2. Vuelta: Toda expresión regular denota un lenguaje regular.

Formalicemos técnicamente este resultado para su posterior prueba:

Teorema ($LR^{\Sigma} \equiv ER^{\Sigma}$)

1. *(Ida) Si $L \in LR^{\Sigma}$, entonces $\exists e \in ER^{\Sigma}$ tal que $L(e) = L$.*
2. *(Vuelta) Si $e \in ER^{\Sigma}$, entonces $\exists L \in LR^{\Sigma}$ tal que $L(e) = L$.*

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Un lenguaje es regular si y solo si es denotado por una expresión regular.

Este resultado implica una ida y una vuelta:

1. Ida: Todo lenguaje regular tiene una expresión regular que lo denota.
2. Vuelta: Toda expresión regular denota un lenguaje regular.

Formalicemos técnicamente este resultado para su posterior prueba:

Teorema ($LR^{\Sigma} \equiv ER^{\Sigma}$)

1. *(Ida) Si $L \in LR^{\Sigma}$, entonces $\exists e \in ER^{\Sigma}$ tal que $L(e) = L$.*
2. *(Vuelta) Si $e \in ER^{\Sigma}$, entonces $\exists L \in LR^{\Sigma}$ tal que $L(e) = L$.*

Solo probaremos el inciso 1, pues 2 es análogo (ejercicio!).

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Un lenguaje es regular si y solo si es denotado por una expresión regular.

Este resultado implica una ida y una vuelta:

1. Ida: Todo lenguaje regular tiene una expresión regular que lo denota.
2. Vuelta: Toda expresión regular denota un lenguaje regular.

Formalicemos técnicamente este resultado para su posterior prueba:

Teorema ($LR^{\Sigma} \equiv ER^{\Sigma}$)

1. *(Ida) Si $L \in LR^{\Sigma}$, entonces $\exists e \in ER^{\Sigma}$ tal que $L(e) = L$.*
2. *(Vuelta) Si $e \in ER^{\Sigma}$, entonces $\exists L \in LR^{\Sigma}$ tal que $L(e) = L$.*

Solo probaremos el inciso 1, pues 2 es análogo (ejercicio!).

<u>Demo 1</u>. Lo probamos por inducción en la forma del lenguaje regular L.

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Casos base:

- Si $L = \emptyset$, entonces tomamos $e = \emptyset$ y $L(\emptyset) = \emptyset$.
- Si $L = \{\epsilon\}$, entonces tomamos $e = \epsilon$ y $L(\epsilon) = \{\epsilon\}$.
- Si $L = \{a\}$, entonces tomamos $e = a$ y $L(a) = \{a\}$.

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Casos base:

- Si $L = \emptyset$, entonces tomamos $e = \emptyset$ y $L(\emptyset) = \emptyset$.
- Si $L = \{\epsilon\}$, entonces tomamos $e = \epsilon$ y $L(\epsilon) = \{\epsilon\}$.
- Si $L = \{a\}$, entonces tomamos $e = a$ y $L(a) = \{a\}$.

Casos Recursivos:

- Si $L = L_1 \cup L_2$, con $L_1, L_2 \in LR^{\Sigma}$, entonces por HI, $\exists e_1, e_2 \in ER^{\Sigma}$ tal que $L(e_1) = L_1$ y $L(e_2) = L_2$. Por lo tanto, podemos tomar $e = e_1 + e_2$, y luego, $L(e_1 + e_2) = L(e_1) \cup L(e_2) = L_1 \cup L_2$.

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Casos base:

- Si $L = \emptyset$, entonces tomamos $e = \emptyset$ y $L(\emptyset) = \emptyset$.
- Si $L = \{\epsilon\}$, entonces tomamos $e = \epsilon$ y $L(\epsilon) = \{\epsilon\}$.
- Si $L = \{a\}$, entonces tomamos $e = a$ y $L(a) = \{a\}$.

Casos Recursivos:

- Si $L = L_1 \cup L_2$, con $L_1, L_2 \in LR^{\Sigma}$, entonces por HI, $\exists e_1, e_2 \in ER^{\Sigma}$ tal que $L(e_1) = L_1$ y $L(e_2) = L_2$. Por lo tanto, podemos tomar $e = e_1 + e_2$, y luego, $L(e_1 + e_2) = L(e_1) \cup L(e_2) = L_1 \cup L_2$.
- Si $L = L_1 L_2$, con $L_1, L_2 \in LR^{\Sigma}$, entonces por HI, $\exists e_1, e_2 \in ER^{\Sigma}$ tal que $L(e_1) = L_1$ y $L(e_2) = L_2$. Por lo tanto, podemos tomar $e = e_1 e_2$, y luego, $L(e_1 e_2) = L(e_1) L(e_2) = L_1 L_2$.

Equivalencia entre Lenguajes Regulares y Expresiones Regulares

Casos base:

- Si $L = \emptyset$, entonces tomamos $e = \emptyset$ y $L(\emptyset) = \emptyset$.
- Si $L = \{\epsilon\}$, entonces tomamos $e = \epsilon$ y $L(\epsilon) = \{\epsilon\}$.
- Si $L = \{a\}$, entonces tomamos $e = a$ y $L(a) = \{a\}$.

Casos Recursivos:

- Si $L = L_1 \cup L_2$, con $L_1, L_2 \in LR^{\Sigma}$, entonces por HI, $\exists e_1, e_2 \in ER^{\Sigma}$ tal que $L(e_1) = L_1$ y $L(e_2) = L_2$. Por lo tanto, podemos tomar $e = e_1 + e_2$, y luego, $L(e_1 + e_2) = L(e_1) \cup L(e_2) = L_1 \cup L_2$.
- Si $L = L_1 L_2$, con $L_1, L_2 \in LR^{\Sigma}$, entonces por HI, $\exists e_1, e_2 \in ER^{\Sigma}$ tal que $L(e_1) = L_1$ y $L(e_2) = L_2$. Por lo tanto, podemos tomar $e = e_1 e_2$, y luego, $L(e_1 e_2) = L(e_1) L(e_2) = L_1 L_2$.
- Si $L = {L_1}^*$, con $L_1 \in LR^{\Sigma}$, entonces por HI, $\exists e_1 \in ER^{\Sigma}$ tal que $L(e_1) = L_1$. Por lo tanto, podemos tomar $e = {e_1}^*$, y luego, $L({e_1}^*) = (L(e_1))^* = {L_1}^*$.

Resumen

- Dimos una definición recursiva de los lenguajes regulares (LR^{Σ}).

Resumen

- Dimos una definición recursiva de los lenguajes regulares (LR^{Σ}).
- Probamos que finitud implica regularidad.

Resumen

- Dimos una definición recursiva de los lenguajes regulares (LR^{Σ}).
- Probamos que finitud implica regularidad.
- Dimos una definición recursiva de las expresiones regulares (ER^{Σ}) y su lenguaje denotado.

Resumen

- Dimos una definición recursiva de los lenguajes regulares (LR^Σ).
- Probamos que finitud implica regularidad.
- Dimos una definición recursiva de las expresiones regulares (ER^Σ) y su lenguaje denotado.
- Probamos que los lenguajes regulares y las expresiones regulares son equivalentes ($LR^\Sigma \equiv ER^\Sigma$).

Resumen

- Dimos una definición recursiva de los lenguajes regulares (LR^{Σ}).
- Probamos que finitud implica regularidad.
- Dimos una definición recursiva de las expresiones regulares (ER^{Σ}) y su lenguaje denotado.
- Probamos que los lenguajes regulares y las expresiones regulares son equivalentes ($LR^{\Sigma} \equiv ER^{\Sigma}$).
- Entonces para probar que un lenguaje es regular, podemos dar directamente una expresión regular que lo denote.

Resumen

- Dimos una definición recursiva de los lenguajes regulares (LR^{Σ}).
- Probamos que finitud implica regularidad.
- Dimos una definición recursiva de las expresiones regulares (ER^{Σ}) y su lenguaje denotado.
- Probamos que los lenguajes regulares y las expresiones regulares son equivalentes ($LR^{\Sigma} \equiv ER^{\Sigma}$).
- Entonces para probar que un lenguaje es regular, podemos dar directamente una expresión regular que lo denote.
- O si, simplemente queremos describir un lenguaje regular, en lugar de utilizar comprensión de conjuntos, podemos escribir directamente su expresión regular equivalente (notar que no es única! existen varias expresiones regulares distintas que denotan un mismo lenguaje).

Bibliografía

Rodrigo De Castro Korgi.
"Teoria de la Computación". Lenguajes, Autómatas, Gramáticas.
Capítulo 1: Alfabetos, cadenas y lenguajes. Sección 1.13 y 1.14.