

# Expresiones regulares

# Recordamos...

Un lenguaje  $L'$  es **regular** si existe un AFD  $M$  que lo acepta. Es decir,  $L(M) = L'$ . De esto concluimos que los lenguajes aceptados por los AFDs forman la familia de los **lenguajes regulares**

Si  $\Sigma = \{a_1, a_2, \dots, a_m\}$  entonces, 1)  $\{a_1\}$ ,  $\{a_2\}$ , .. son lenguajes regulares. 2)  $\{\}$  es un lenguaje regular y 3) el conjunto que solo tiene a  $\lambda$ , es también un lenguaje regular

También sabemos que dado dos lenguajes regulares  $L_1$  y  $L_2$ ,

La **unión**, **concatenación**, la cláusula de **Kleene**, la operación de **reversa**, el **complemento** y la **intersección** preservan la propiedad de ser Lenguajes regulares

# Definición inductiva de Lenguajes regulares

Sea  $\Sigma = \{a_1, a_2, \dots, a_m\}$ , un lenguaje regular sobre  $\Sigma$  es cualquier conjunto que pueda formarse por una secuencia finita de aplicaciones de las siguientes reglas:

- $\{a_1\}, \{a_2\}, \dots, \{a_m\}$  son lenguajes regulares
- $\{\}$  es un lenguaje regular
- el conjunto que solo contiene a  $\lambda$  es un lenguaje regular
- Dado  $L_1$  y  $L_2$  regulares,  $L_1 L_2$  es un lenguaje regular
- Dado  $L_1$  y  $L_2$  regulares,  $L_1 \cup L_2$  es un lenguaje regular
- Dado  $L_1$  y  $L_2$  regulares,  $L_1 \cap L_2$  es un lenguaje regular
- Dado  $L_1$  regular,  $L_1^*$  es un lenguaje regular

# Expresiones regulares

Dado un alfabeto  $\Sigma$

- $\emptyset$  ( el conjunto vacío ),  $\epsilon$  ( epsilon ) y cualquier  $a \in \Sigma$ , son expresiones regulares
- sea  $r_1$  y  $r_2$ , dos expresiones regulares:
  - $r_1 + r_2$  ( **suma** ) es una expresión regular
  - $r_1 \cdot r_2$  ( **concatenación** ) es una expresión regular
  - $r_1^*$  ( **cláusula de kleene** ) es una expresión regular
  - $(r_1)$  ( **paréntesis** ) es una expresión regular

Ejemplos:

$$\begin{array}{ll} (a+b \cdot c)^* \cdot (c + \emptyset) & (a+b)^* \cdot (a+b \cdot b) \\ (a+b) \cdot a^* & (0+1)^* \cdot 0 \cdot 0 \cdot (0+1)^* \end{array}$$

Esto **NO** es una RE

↓

$$(a + b +)$$

# Función de interpretación

Notar que necesitamos alguna forma de interpretar una expresión  $r$ . La llamaremos **función de interpretación**,  $L(r)$

- **Casos primitivos**

- $L(\emptyset) = \{ \}$
- $L(\epsilon) = \{ \lambda \}$
- $L(a) = \{ a \}$  para cualquier  $a \in \Sigma$

- **Casos inductivos**

- $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
- $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$
- $L(r_1^*) = (L(r_1))^*$
- $L((r_1)) = L(r_1)$

$$\begin{aligned}
 L((a + b) \cdot a^*) &\Rightarrow L(a + b) \cdot L(a^*) \Rightarrow L(a) \cup L(b) \cdot (L(a))^* \Rightarrow \\
 &\Rightarrow (\{a\} \cup \{b\}) \cdot (\{a\})^* \Rightarrow \{a, b\} \cdot \{\lambda, a, aa, aaa, \dots\} \Rightarrow \{a, aa, aaa, \dots, b, ba, baaa, \dots\}
 \end{aligned}$$

Notar que las expresiones regulares describen **Lenguajes regulares**

Ejemplos

$$r_1 = (a \cdot a)^* \cdot (a \cdot b)^* \cdot b$$

$$L(r_1) = \{a^{2n} b^{2m} b, n, m \geq 0\}$$

$$r_2 = (1 + (0 \cdot 1))^* \cdot (0 + \epsilon)$$

$$L(r_2) = \{x \in \{0, 1\}^* : |x|_{00} = 0\}$$

# Equivalencia entre expresiones regulares

Sea  $r_1$  y  $r_2$ , dos expresiones regulares,  $r_1$  y  $r_2$  son equivalentes si  $L(r_1) = L(r_2)$

Ejemplo:  $r_1 = (1 + (0 \cdot 1))^* \cdot (0 + \varepsilon)$  y  $r_2 = ((1^* \cdot (0 \cdot 1)^*)^* \cdot (0 + \varepsilon)) + (1^* \cdot (0 + \varepsilon))$  son **equivalentes**

Para probar que dos expresiones regulares son equivalentes debemos mostrar que describen el mismo lenguaje, **buscamos otra manera...**

# Propiedades algebraicas de las expresiones regulares

Sean tres expresiones regulares  $r_1$ ,  $r_2$  y  $r_3$  entonces:

- **Asociatividad:**  $(r_1 \cup r_2) \cup r_3 = r_1 \cup (r_2 \cup r_3)$  o bien  $(r_1 \cdot r_2) \cdot r_3 = r_1 \cdot (r_2 \cdot r_3)$
- **Conmutatividad:**  $r_1 \cup r_2 = r_2 \cup r_1$  PERO  $r_1 \cdot r_2 \neq r_2 \cdot r_1$
- **Distributividad:**  $r_1 \cdot (r_2 \cup r_3) = r_1 r_2 \cup r_1 r_3$  o bien  $(r_1 \cup r_2) \cdot r_3 = r_1 r_2 \cup r_1 r_3$
- **Neutro:**  $r_1 \cdot \lambda = \lambda \cdot r_1 = r_1$
- **Absorbente:**  $r_1 \cdot \emptyset = \emptyset \cdot r_1 = \emptyset$
- **Neutro de U:**  $r_1 \cup \emptyset = \emptyset \cup r_1 = r_1$



## Aclaración

- A fin de facilitar la lectura y escritura de las ER, no escribimos los "." y los paréntesis. Asumimos la **precedencia**:  $+ < * < . < U$

$$\text{Ej: } ((1^* \cdot (0 \cdot 1)^*)^* \cdot (0 + \varepsilon)) + (1^* \cdot (0 + \varepsilon)) \longrightarrow (1^* (01)^*)^* (0 + \varepsilon) + 1^* (0 + \varepsilon)$$

**Definición:**  $r \subseteq s \iff r + s = s$

**Propiedad:**  $r + s = t \Rightarrow r + t = t \wedge s + t = t$

**Teorema:**  $r \subseteq s \Rightarrow r^* \subseteq s^*$

# Expresiones y Lenguajes Regulares

**Teorema:** Los lenguajes generados por expresiones regulares es exactamente la familia de los lenguajes regulares

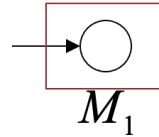
**Demo:** Mostramos la doble inclusión

1.  $\{ \text{Lenguajes generados por ER} \} \subseteq \{ \text{Lenguajes regulares} \}$
2.  $\{ \text{Lenguajes regulares} \} \subseteq \{ \text{Lenguajes generados por ER} \}$

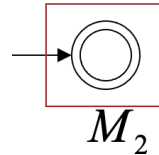
**Parte 1:** Para cualquier expresión regular  $r$ , el lenguaje  $L(r)$  es regular.

**Prueba:** Por inducción sobre el tamaño de  $r$

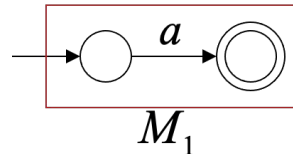
- Casos base,  $\emptyset$ ,  $\varepsilon$ ,  $a \in \Sigma$



$$L(M_1) = \{\} = L(\emptyset)$$



$$L(M_2) = \{\lambda\} = L(\varepsilon)$$



$$L(M_3) = \{a\} = L(a)$$

lenguajes  
regulares

**Parte 1:** Para cualquier expresión regular  $r$ , el lenguaje  $L(r)$  es regular.

**Prueba:** Por inducción sobre el tamaño de  $r$

- Casos inductivos,  $r_1 + r_2$ ,  $r_1 \cdot r_2$ ,  $r_1^*$  y  $((r_1))$

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1) L(r_2)$$

$$L(r_1^*) = (L(r_1))^*$$

$$L((r_1)) = L(r_1)$$

**Parte 1:** Para cualquier expresión regular  $r$ , el lenguaje  $L(r)$  es regular.

**Prueba:** Por inducción sobre el tamaño de  $r$

- Casos inductivos,  $r_1 + r_2$ ,  $r_1 \cdot r_2$ ,  $r_1^*$  y  $((r_1))$

**Por HI:**  $L(r_1)$  y  $L(r_2)$  son lenguajes regulares

También sabemos que los lenguajes regulares son cerrados bajo la unión, la concatenación y la clausura

$$\begin{aligned} &L(r_1) \cup L(r_2) \\ &L(r_1) L(r_2) \\ &(L(r_1))^* \end{aligned}$$

**Parte 1:** Para cualquier expresión regular  $r$ , el lenguaje  $L(r)$  es regular.

**Prueba:** Por inducción sobre el tamaño de  $r$

- Casos inductivos,  $r_1 + r_2$ ,  $r_1 \cdot r_2$ ,  $r_1^*$  y  $((r_1))$

$$L(r_1 + r_2) = L(r_1) \cup L(r_2)$$

$$L(r_1 \cdot r_2) = L(r_1) L(r_2)$$

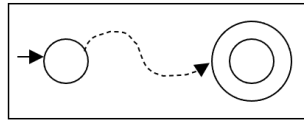
$$L(r_1^*) = (L(r_1))^*$$

Son lenguajes  
regulares

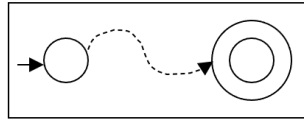
Usando las propiedades de clausura de estas operaciones, podemos construir recursivamente el AFND  $M$  que acepta  $L(M) = L(r)$

**Ejemplo:**  $r = r_1 + r_2$

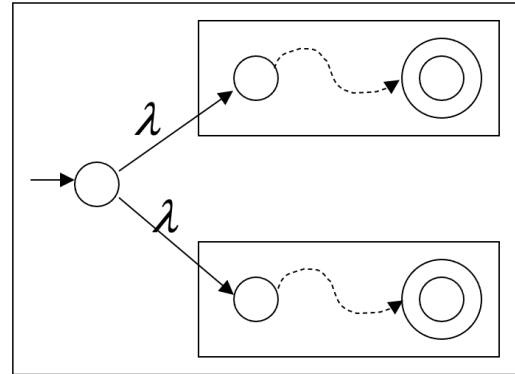
$L(M_1) = L(r_1)$



$L(M_2) = L(r_2)$



$L(M) = L(r)$

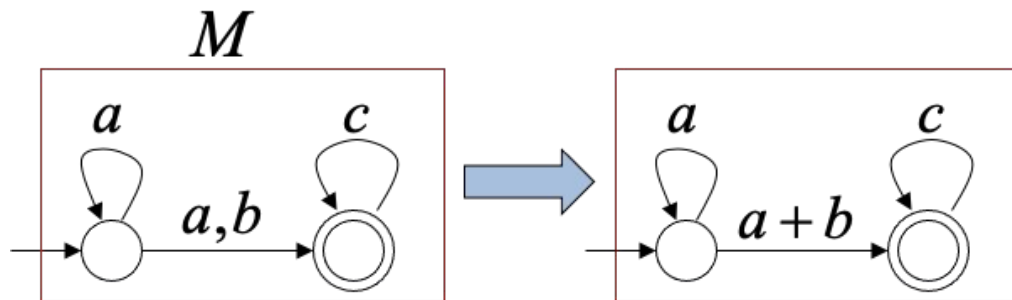


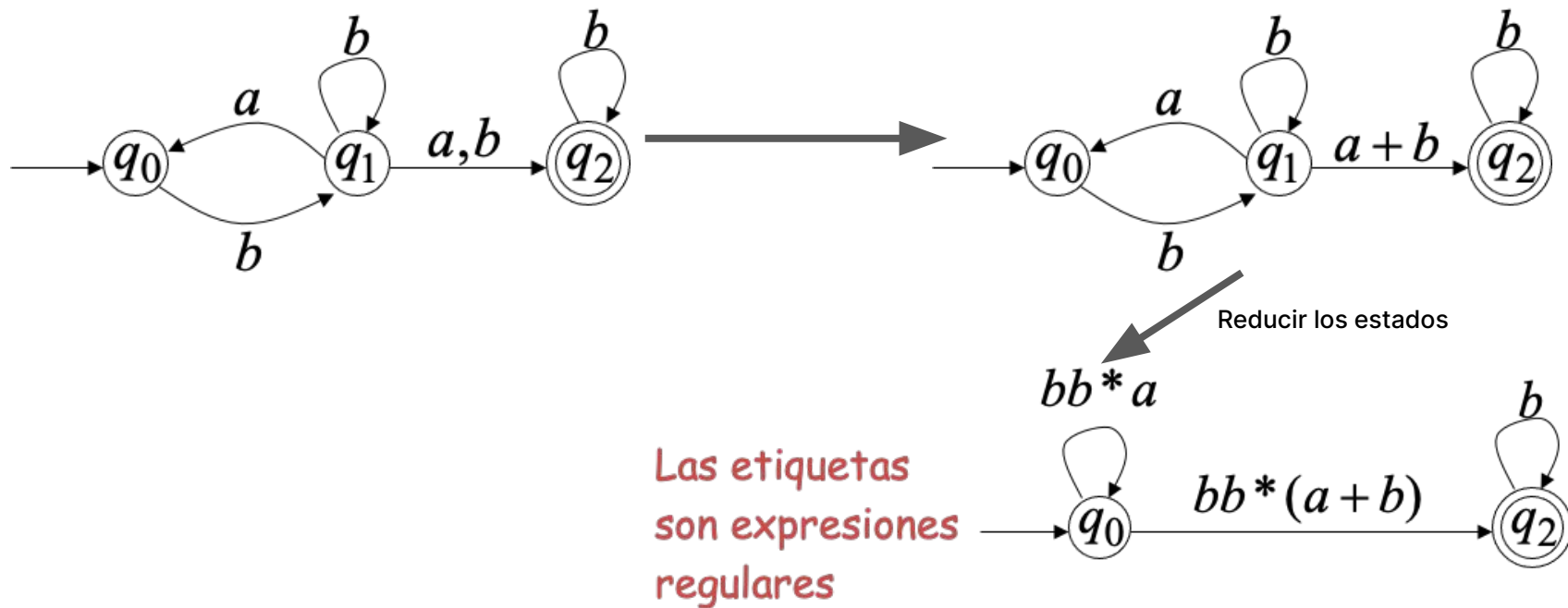


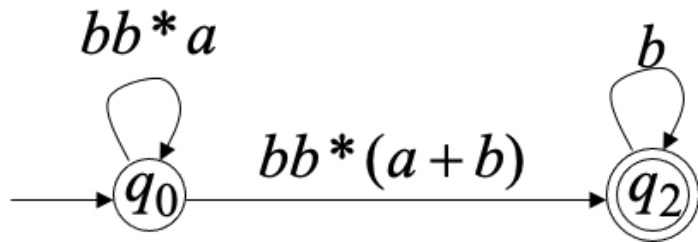
**Parte 2:** Para cualquier lenguaje regular  $L$ , existe una expresión regular  $r$  con  $L(r) = L$

**Prueba:** Convertimos un AFND que acepta  $L$  a una expresión regular

- Dado que  $L$  es regular, existe un AFND  $M$  que lo acepta
- A partir de  $M$  construimos el grafo de transición generalizado equivalente en el cual las etiquetas de transición son expresiones regulares



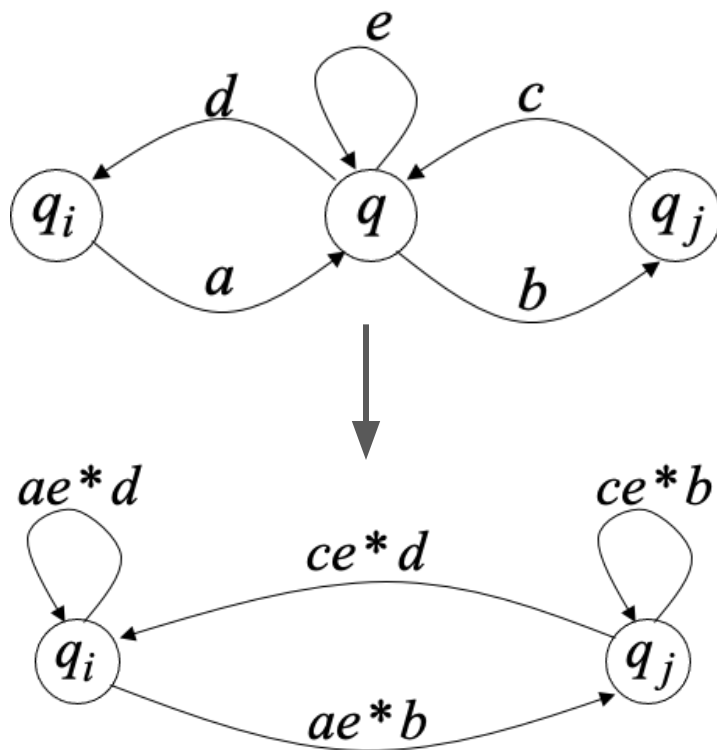




$$r = (bb^*a)^*bb^*(a+b)b^*$$

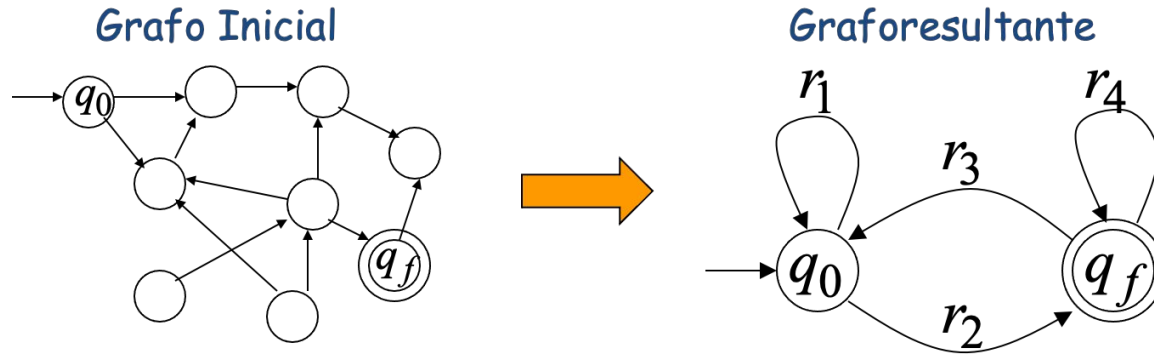
$$L(r) = L(M) = L$$

# En general



# En general

Repitiendo el proceso hasta que queden dos estados, el grafo resultante es



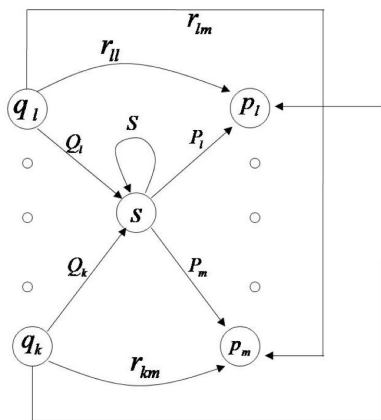
$$r = r_1 * r_2 (r_4 + r_3 r_1 * r_2)^*$$

$$L(r) = L(M) = L$$

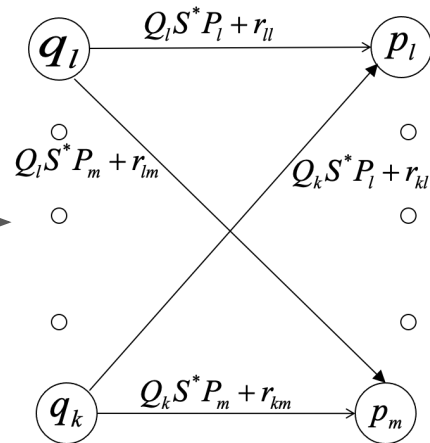
# Método de eliminación de estados

Supongamos que deseamos eliminar el estado  $s$

- Se eliminan todos los arcos que incluyen a " $s$ "
- Se introducen, para cada predecesor  $q_i$  de  $s$  y cada sucesor  $p_j$  de  $s$ , una expresión regular que representa todas las rutas que inician en  $q_i$ , van a  $s$ , quizás hacen un loop en  $s$  (cero o más veces, y finalmente van a  $p_j$ . La expresión para estas rutas es  $Q_i S^* P_j$ . Esta expresión se suma al arco que va de  $q_i$  a  $p_j$ . Si este arco no existe, se añade primero uno con la expresión  $\emptyset$



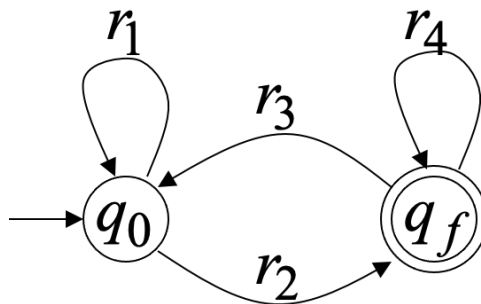
$r_{kl}$



# Estrategia para construir una RE equivalente

1 - Para cada estado final  $q_f$ , aplicar el proceso de reducción para producir un autómata equivalente con expresiones regulares como etiquetas en los arcos. Eliminar todos los estados excepto  $q_f$  y el estado inicial  $q_0$ .

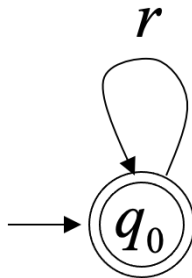
2 - Si  $q_f \neq q_0$ , se genera un autómata con 2 estados como el siguiente,



$$r = r_1 * r_2 (r_4 + r_3 r_1 * r_2)^* = (r_1 + r_2 r_4 * r_3)^* r_2 r_4 *$$

# Estrategia para construir una RE equivalente

3 - Si el estado inicial es un estado final, también se debe hacer una eliminación de estados del autómata original que elimine todos los estados menos el inicial y dejamos un autómata como el siguiente:

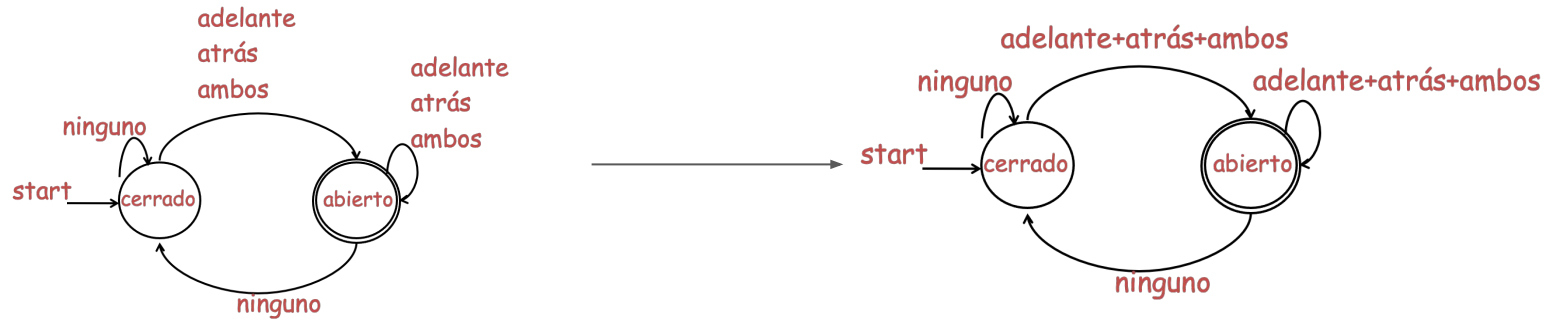


4 - La expresión final es la suma de todas las expresiones derivadas del autómata reducido para cada estado de aceptación por las reglas 2 y 3



# Ejemplo: Puerta

Transformamos el AFND del problema de la puerta en una RE

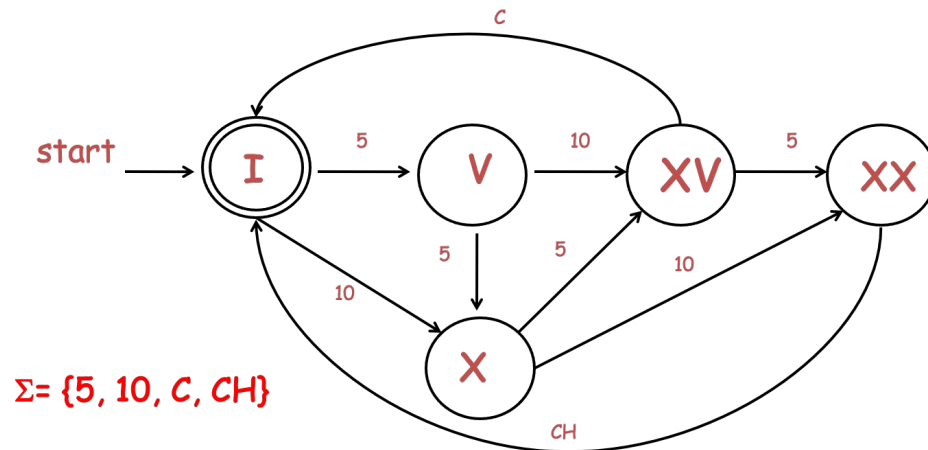


$\text{ninguno}^* (\text{adelante+atrás+ambos}) ((\text{adelante+atrás+ambos}) + \text{ninguno} \text{ninguno}^* (\text{adelante+atrás+ambos}))^* =$

$(\text{ninguno}(\text{adelante+atrás+ambos})(\text{adelante+atrás+ambos})^* \text{ninguno})^* (\text{adelante+atrás+ambos})(\text{adelante+atrás+ambos})^*$

# Ejemplo 2: Máquina expendedora

Transformamos el AFND del problema de la máquina expendedora en una RE



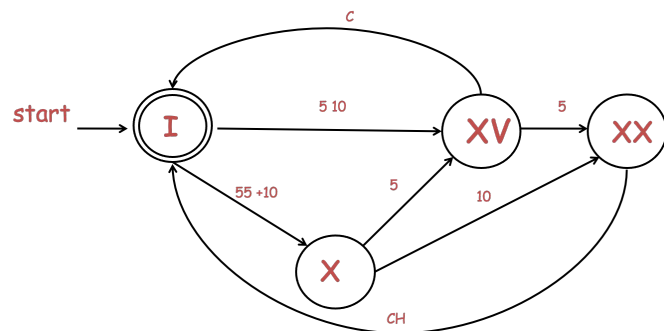
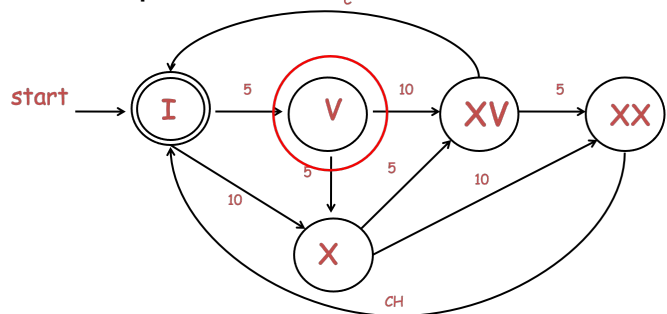
$\Sigma = \{5, 10, C, CH\}$

$L_0 = \{\lambda, 510C, 555C, 105C, 5105CH, 5555CH, 1055CH, 5510CH, 1010CH\}$

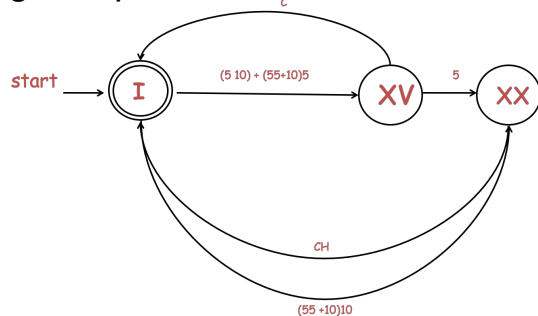
$L(M) = L_0^*$

# Ejemplo 2: Máquina expendedora

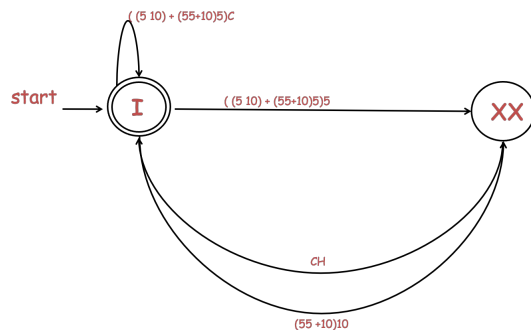
Primer paso: Eliminar estado V



Segundo paso: Eliminar estado X

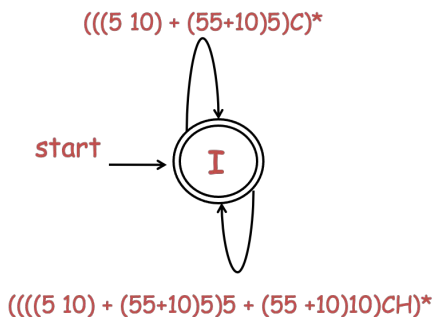
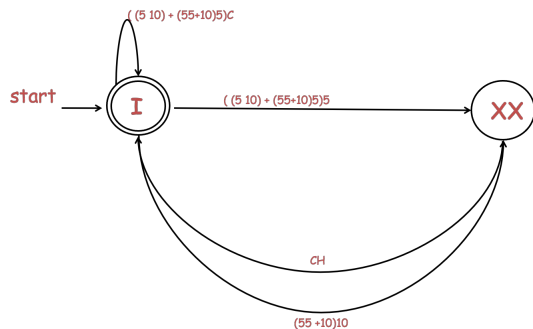


Tercer paso: Eliminar estado XV

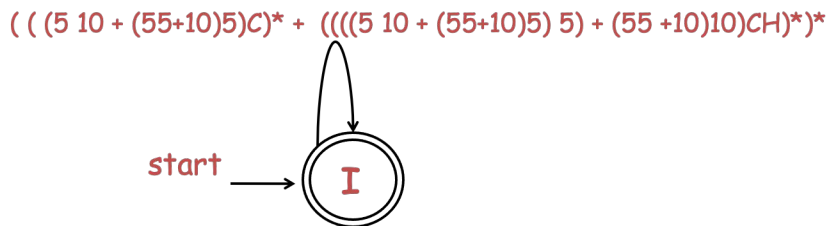


# Ejemplo 2: Máquina expendedora

Cuarto paso: Eliminar estado XX



Paso final

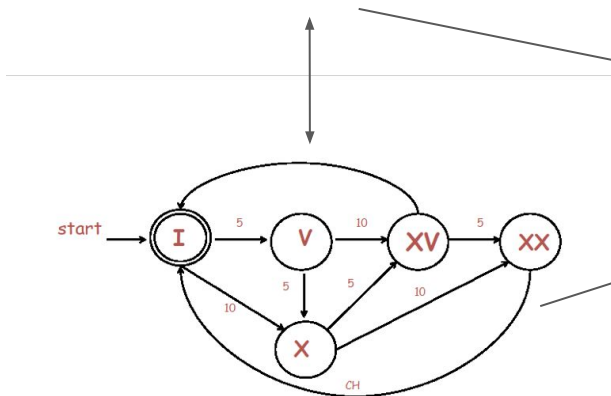


Vemos que:  $r = (((((510) + (55+10)5)C)^* + (((((510 + (55+10)5)5 + (55 +10)10)CH)^*)^*)^* = L(r) = L_0^*$

# Observación

## Sintaxis

$r = (((510) + (55+10)5)C)^* + (((510 + (55+10)5)5 + (55 + 10)10)CH)^* )^*$



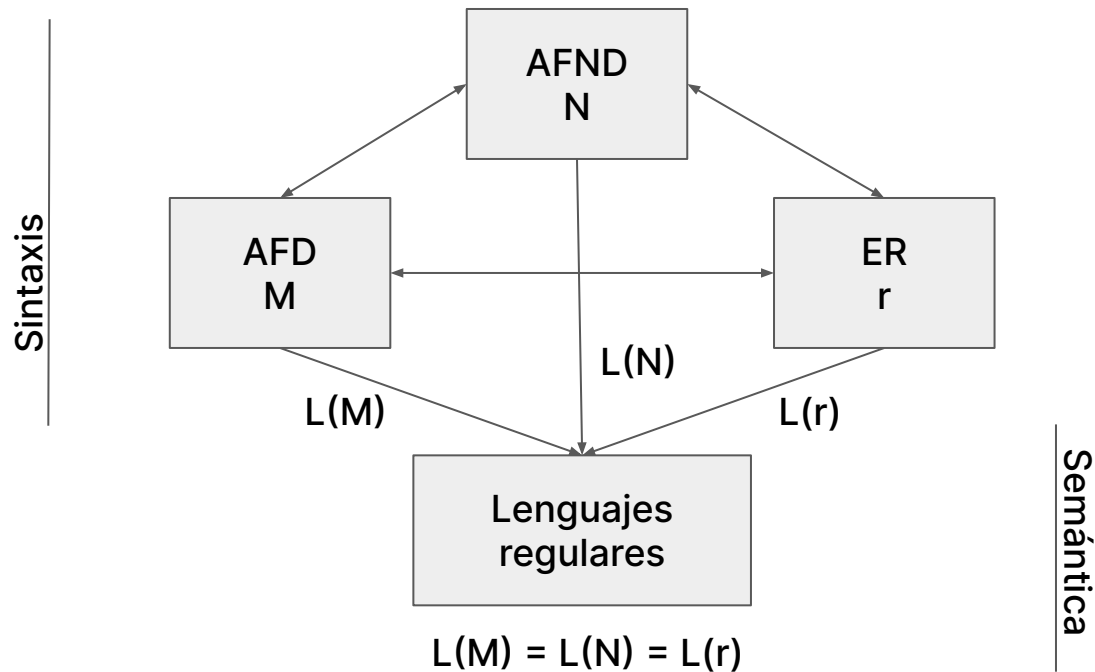
## Semántica

$L_0^*$

L

L

# Representación estándar de los Lenguajes Regulares



**Nota:** Cuando decimos “*Tenemos un lenguaje regular  $L$* ” nos referimos a “*El lenguaje  $L$  está en una representación estándar*” ( AFD, AFND, ER )