

Capítulo 7: Expresiones Regulares

7.1. Concepto de expresión regular

7.1.1. Definición

7.1.2. Lenguaje descrito

7.1.3. Propiedades

7.2. Teoremas de equivalencia

7.2.1. Obtener un AFND a partir de una expresión regular

7.2.2. Obtener una expresión equivalente a partir de un autómata finito

7.1. Concepto de Expresión Regular

El objetivo de las expresiones regulares es representar todos los posibles lenguajes definidos sobre un alfabeto Σ , en base a una serie de lenguajes primitivos, y unos operadores de composición.

Lenguajes primitivos: **el lenguaje vacío, el lenguaje formado por la palabra vacía, y los lenguajes correspondientes a los distintos símbolos del alfabeto.**

Operadores de composición: **la unión, la concatenación y el cierre.**

Ejemplo:

1. Lenguaje formado por las cadenas que terminan en 01:

$$\{0,1\}^*.\{01\}=$$

$$(\{0\} \cup \{1\})^*.\{01\}$$

$$\Rightarrow \text{Expresión regular: } (0+1)^*01$$

2. Lenguaje formado por palabras de longitud par sobre a's y b's:

$$\{aa,ab,ba,bb\}^*=$$

$$(\{aa\} \cup \{ab\} \cup \{ba\} \cup \{bb\})^*$$

$$\Rightarrow \text{Expresión: } (aa+ab+ba+bb)^*$$

7.1.1 Definición

Dado un alfabeto Σ , las *expresiones regulares sobre Σ* se definen de forma recursiva por las siguientes reglas:

1. Las siguientes expresiones son expresiones regulares primitivas:
 - \emptyset
 - λ
 - a , siendo $a \in \Sigma$.
2. Sean α y β expresiones regulares, entonces son expresiones regulares derivadas:
 - $\alpha + \beta$ (unión)
 - $\alpha.\beta$ (o simplemente $\alpha\beta$) (concatenación)
 - α^* (cierre)
 - (α)
3. No hay más expresiones regulares sobre Σ que las construidas mediante estas reglas.

Precedencia de los operadores:

1. $()$
2. $*$ cierre
3. $.$ concatenación
4. $+$ unión

Ejemplo:

Algunos ejemplos de expresión regular son:

$(0 + 1)^*01$
 $(aa + ab + ba + bb)^*$
 $a^*(a + b)$
 $(aa)^*(bb)^*b$

7.1.2 Lenguaje descrito por una ER

Definición (Lenguaje descrito por una ER):

Sea r una expresión regular sobre Σ . El *lenguaje descrito por r* , $L(r)$, se define recursivamente de la siguiente forma:

1. Si $r = \emptyset \Rightarrow L(\emptyset) = \emptyset$
2. Si $r = \lambda \Rightarrow L(\lambda) = \{\lambda\}$
3. Si $R = a, a \in \Sigma \Rightarrow L(a) = \{a\}$
4. Si $R = \alpha + \beta \Rightarrow L(\alpha + \beta) = L(\alpha) \cup L(\beta)$
5. Si $R = \alpha.\beta \Rightarrow L(\alpha.\beta) = L(\alpha)L(\beta)$
6. Si $R = \alpha^* \Rightarrow L(\alpha^*) = L(\alpha)^*$
7. Si $R = (\alpha) \Rightarrow L((\alpha)) = L(\alpha)$

donde α y β son expresiones regulares.

Ejemplo: Mostrar el lenguaje descrito por una ER mediante notación conjuntista:

$$\begin{aligned}
 1. L(a^*(a+b)) &= L(a^*)L((a+b)) = L(a)^*L(a+b) \\
 &= L(a)^*(L(a) \cup L(b)) = \{a\}^*(\{a\} \cup \{b\}) \\
 &= \{\lambda, a, aa, aaa, \dots\} \{a, b\} \\
 &= \{a, aa, \dots, b, ab, aab, \dots\} \\
 &= \{a^n | n \geq 1\} \cup \{a^n b | n \geq 0\}
 \end{aligned}$$

$$2. L((aa)^*(bb)^*b) = \{a^{2n}b^{2m+1} | n, m \geq 0\}$$

$$3. \text{ Si } \Sigma = \{a, b, c\}, \text{ entonces } L((a+b+c)^*) = \Sigma^*.$$

$$4. L(a^*. (b+c))$$

$$5. L(0^*. 1. 0^*)$$

$$6. L((a+b+c+\dots+z)^*. (a+b)^*)$$

$$7. \text{ ¿Que lenguaje describe la expresión } a^*. (a+b). c^*?$$

$$8. \text{ Dado el lenguaje } L = \{w \mid w \in \{a, b, c\}^* \text{ donde } w \text{ tiene al menos un par de } a\text{'s consecutivos}\}. \text{ Escribe la expresión regular para } L.$$

$$9. \text{ Escribe todas las palabras de longitud } < 4 \text{ de } L((a+b)^*. b. (a+ab)^*).$$

7.1.3 Propiedades de Expresiones Regulares

Definición (equivalencia de ER):

Dos expresiones regulares r_1 y r_2 se dicen *equivalentes*, $r_1 = r_2$, si describen el mismo lenguaje, esto es, si $L(r_1)=L(r_2)$.

En base a esta definición se pueden establecer las siguientes equivalencias y propiedades:

Respecto a las operaciones $+$ y $.$:

1. $+$ y $.$ son asociativas: $\alpha+(\beta+\gamma)=(\alpha+\beta)+\gamma=\alpha+\beta+\gamma$ y $\alpha.(\beta.\gamma)=(\alpha.\beta).\gamma=\alpha.\beta.\gamma$
2. $+$ es conmutativa e idempotente: $\alpha+\beta=\beta+\alpha$ y $\alpha+\alpha=\alpha$
3. Distributividad: $\alpha.(\beta+\gamma)=\alpha.\beta+\alpha.\gamma$ y $(\alpha+\beta).\gamma=\alpha.\gamma+\beta.\gamma$
4. Elemento neutro: $\alpha.\lambda=\lambda.\alpha=\alpha$ y $\alpha+\emptyset=\emptyset+\alpha=\alpha$
5. $\emptyset.\alpha=\alpha.\emptyset=\emptyset$
6. Si $\lambda \in L(\alpha)$, entonces $\alpha+\lambda=\alpha$

Respecto a la operación $*$:

7. $\alpha^*=\lambda+\alpha.\alpha^*$
8. $\lambda^*=\lambda$
9. $\emptyset^*=\lambda$
10. $\alpha^*.\alpha^*=\alpha^*$
11. $\alpha.\alpha^*=\alpha^*.\alpha$
12. $(\alpha^*)^*=\alpha^*$
13. $(\alpha^*+\beta^*)^*=(\alpha^*.\beta^*)^*=(\alpha+\beta)^*=(\alpha^*.\beta)^*.\alpha^*$
14. $(\alpha.\beta)^*.\alpha=\alpha.(\beta.\alpha)^*$

Para comprobar si dos expresiones son equivalentes se puede intentar transformarlos mediante estas reglas en una misma expresión.

Ejemplos:

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

1. $c^*.c+c^*=c^*$ ¿?

$$\begin{aligned}c^*.c+c^* &= c^*.c+c^*+\lambda && \text{(por 6)} \\&= c.c^*+c^*+\lambda && \text{(por 11)} \\&= \lambda+c.c^*+c^* && \text{(por 2)} \\&= c^*+c^* && \text{(por 7)} \\&= c^* && \text{(por 2)}\end{aligned}$$

2. $c+c^*=c^*$ ¿?

$$\begin{aligned}c+c^* &= c+\lambda+c.c^* && \text{(por 7)} \\&= \lambda+c+c.c^* && \text{(por 2)} \\&= \lambda+c.\lambda+c.c^* && \text{(por 4)} \\&= \lambda+c.(\lambda+c^*) && \text{(por 3)} \\&= \lambda+c.c^* && \text{(por 6)} \\&= c^* && \text{(por 7)}\end{aligned}$$

3. $((c+b.a)^*.a^*)^*=((c+b.a)+a)^*$ ¿?

4. Dado dos expresiones regulares $R = b.c+a.c^*.a.c+a.c^*.c+a$ y $S = (b+a.c^*a).c+a.c^*$. ¿Representan S y R el mismo lenguaje?

5. Demuestre que las expresiones $R = (a^*. (b+c)^* + b^*)^*$ y $S = (a+b+c)^*$ son iguales.

Observación: De este modo sólo se puede demostrar que dos expresiones regulares son equivalentes. Sin embargo, mediante este método, no es posible demostrar que dos expresiones regulares describen lenguajes distintos.

7.2 Teoremas de equivalencia

- Tal y como indica su nombre, mediante expresiones regulares se puede representar lenguajes regulares. De hecho, la clase de lenguajes que se puede representar mediante una ER, es equivalente a la clase de lenguajes regulares.
- Hasta ahora hemos visto que los lenguajes regulares pueden describirse mediante:
 - Gramáticas lineales por la izquierda
 - Gramáticas lineales por la derecha
 - Autómatas finitos deterministas
 - Autómatas finitos no deterministas
- Por tanto, deben existir algoritmos que permiten obtener un autómata o una gramática regular a partir de una expresión regular y viceversa.

7.2.1 ER equivalente a un autómata finito

Tres métodos principales para convertir expresiones regulares en autómatas:

- Método de las r^n_{ij} (Hopcroft).
- Eliminación de estados (Hopcroft, Linz)
- Ecuaciones características (Alfonseca, Isasi) (equivalente al método de la eliminación de estados)

Definición (ecuación característica):

Sea un autómata finito $A=(\{q_0, q_1, \dots, q_n\}, \Sigma, f, q_0, F)$.

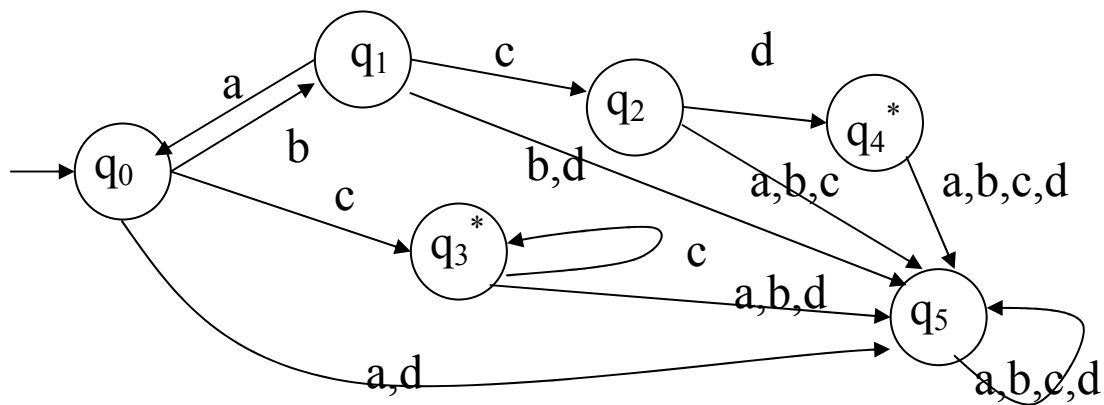
Cada estado del autómata tiene asignado una ecuación característica correspondiente, que describe las distintas formas de llegar desde este estado a un estado final. La ecuación característica para el estado q_i es la siguiente:

$$X_i = b_j X_j + b_k X_k + \dots + b_w X_w + a_i$$

donde:

- La expresión $b_k X_k$ forma parte de la ecuación **si y sólo si** existe una transición del estado q_i al estado q_k para el símbolo de entrada b_k
- a_i es una expresión tal que $a_i = \lambda$ si $q_i \in F$; $a_i = \emptyset$ en otro caso.

Ejemplo:



$$x_0 = b.x_1 + c.x_3 + a.x_5 + d.x_5 + \emptyset = b.x_1 + c.x_3 + (a+d).x_5 + \emptyset$$

$$x_1 = c.x_2 + a.x_0 + (b+d).x_5 + \emptyset$$

$$x_2 = d.x_4 + (a+b+c).x_5 + \emptyset$$

$$x_3 = c.x_3 + (a+b+d).x_5 + \lambda$$

$$x_4 = (a+b+c+d).x_5 + \lambda$$

$$x_5 = (a+b+c+d).x_5 + \emptyset$$

Observación:

1. Se puede definir las ecuaciones características para autómatas finitos deterministas y no deterministas.
2. X_i es una expresión regular (con variables) que describe las cadenas que llevan del estado q_i a un estado final. Evidentemente $L(X_0) = L(A)$.

Teniendo todas las ecuaciones características, se puede resolver la ecuación para el estado inicial obteniendo la expresión regular del lenguaje. El siguiente lema proporciona una regla para eliminar las variables en las ecuaciones.

Lema 1:

Sea X una variable y A y B expresiones regulares. Si $X=A.X+B$ y $\lambda \notin L(A)$, entonces $X=A^*.B$.

Demostración:

Sea cualquier palabra $x \in L(X)$ con $|x|=n$ y $X=A.X+B$. Se demuestra que se cumple $x \in L(A^*.B)$. Consideramos la definición de X :

$$X=A.X+B$$

$$X=A.(A.X+B)+B=A^2.X+A.B+B$$

$$X=A^2.(A.X+B)+A.B+B=A^3.X+A^2.B+A.B+B$$

...

$$X=A^{n+1}.X+A^n.B+\dots+A.B+B=A^{n+1}.X+(A^n+\dots+A+\lambda).B$$

- Por tanto, $L(X)=L(A^{n+1}.X) \cup L((A^n+\dots+A+\lambda).B)$ y dado que $x \in L(X)$ se sigue $x \in L(A^{n+1}.X)$ o $x \in L((A^n+\dots+A+\lambda).B)$.
- Dado que $\lambda \notin L(A)$, para cualquier $w \in L(A^{n+1}.X)$ se cumple $|w| \geq n+1$.
- x tiene longitud n por lo que $x \notin L(A^{n+1}.X)$.
- Por tanto, $x \in L((A^n+\dots+A+\lambda).B)$ y, entonces también se verifica $x \in L(A^*.B)$.

Razonando de forma similar se puede demostrar que para cualquier palabra $x \in L(A^*.B)$ también se cumple $x \in L(X)$.

Ejemplos: Resolución de ecuaciones

1. $X=abX$ $X=abX+\emptyset=(ab)^*\emptyset=\emptyset$
2. $X=abX+\lambda$ $X=(ab)^*\lambda=(ab)^*$
3. $X=abX+cX$ $X=(ab+c)X+\emptyset=(ab+c)^*\emptyset=\emptyset$

Teorema 1:

Dado un autómata finito $A=(Q,\Sigma,f,q_0,F)$, existe una expresión regular R tal que $L(R)=L(A)$.

Demostración:

La expresión regular equivalente se obtiene resolviendo de forma sucesiva las ecuaciones características del autómata. La expresión regular R será la que se obtiene a partir de la ecuación característica correspondiente al estado inicial del autómata: $R=X_0$.

Ejemplo: (para el autómata anterior)

Ecuaciones características:

$$x_0=b.x_1+c.x_3+(a+d).x_5+\emptyset$$

$$x_1=c.x_2+a.x_0+(b+d).x_5+\emptyset$$

$$x_2=d.x_4+(a+b+c).x_5+\emptyset$$

$$x_3=c.x_3+(a+b+d).x_5+\lambda$$

$$x_4=(a+b+c+d).x_5+\lambda$$

$$x_5=(a+b+c+d).x_5+\emptyset$$

$$\text{Resolviendo } x_5: x_5=(a+b+c+d)^*.\emptyset=\emptyset$$

$$\text{Resolviendo } x_4: x_4=(a+b+c+d).\emptyset+\lambda=\lambda$$

$$\text{Resolviendo } x_2: x_2=d.\lambda+(a+b+c).\emptyset+\emptyset=d$$

$$\text{Resolviendo } x_3: x_3=c.x_3+(a+b+d).\emptyset+\lambda=c.x_3+\lambda=c^*.\lambda=c^*$$

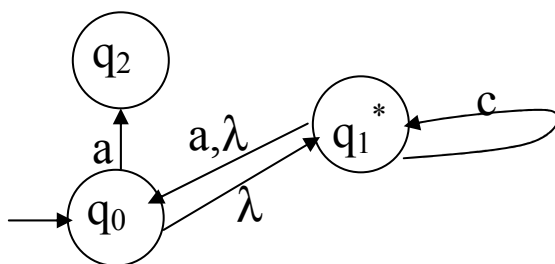
$$\text{Resolviendo } x_1: x_1=c.d+a.x_0+(b+d).\emptyset+\emptyset=cd+a.x_0$$

$$\begin{aligned}\text{Resolviendo } x_0: x_0 &= b.(cd+a.x_0)+c.c^*+(a+d).\emptyset+\emptyset \\ &= bax_0+bcd+c.c^*=(ba)^*(bcd+cc^*)\end{aligned}$$

Observaciones:

- La aplicación de la regla “Si $X=A.X+B$, entonces $X=A^*.B$ ” sólo es posible si $\lambda \notin L(A)$.
- Si el autómata tiene transiciones λ , entonces es posible que **no se puede aplicar esta regla**.
- En consecuencia, será necesario eliminar transiciones λ antes (convertir el autómata en uno sin transiciones λ).

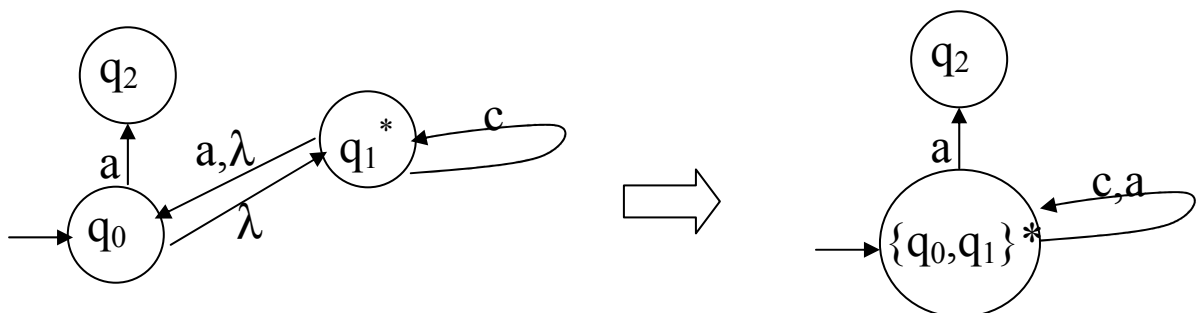
Ejemplo:



$$\begin{aligned}
 x_2 &= \emptyset \\
 x_1 &= cx_1 + (a + \lambda)x_0 + \lambda \\
 &= c^* \cdot ((a + \lambda)x_0 + \lambda) \\
 &= (c^* \cdot a + c^*)x_0 + c^* \\
 x_0 &= \lambda x_1 + ax_2 + \emptyset \\
 &= \lambda((c^* \cdot a + c^*)x_0 + c^*) + a\emptyset \\
 &= (c^* \cdot a + c^*)x_0 + c^* \\
 &\text{!!!} \neq (c^* \cdot a + c^*)^* \cdot c^* \text{ !!!!}
 \end{aligned}$$

Transformación del autómata (eliminación de transiciones λ):

- similar a la conversión a un autómata determinista
- los estados nuevos son las clausuras de los estados originales respecto a λ :



$$\begin{aligned}
 x_2 &= \emptyset \\
 x_{01} &= (c+a)x_{01} + \lambda + ax_2 \\
 &= (c+a)x_{01} + \lambda + a\emptyset = (c+a)x_{01} + \lambda = (c+a)^* \cdot \lambda = (c+a)^*
 \end{aligned}$$

Ejemplos: Obtener las expresiones regulares para los siguientes autómatas:

1.

A		a	b
\rightarrow	p	q	p
*	q	p	q

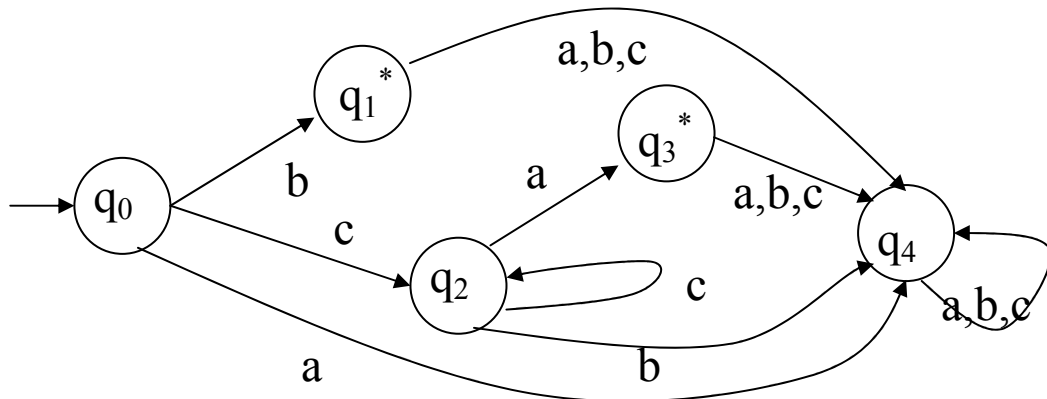
$$\begin{aligned}x_p &= ax_q + bx_p + \emptyset \\x_q &= ax_p + bx_q + \lambda \\ \text{Solución: } x_p &= (ab^*a + b)^*(ab^*)\end{aligned}$$

2.

A		0	1
\rightarrow	A	B	A
*	B	C	A
	C	C	C

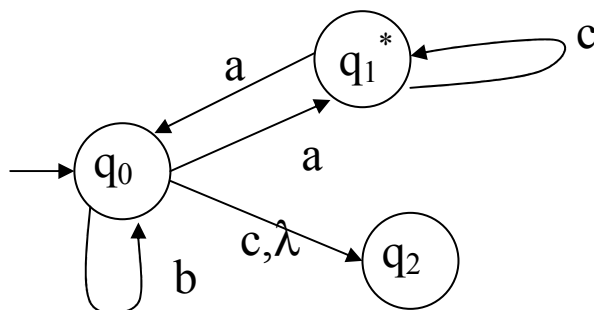
$$\begin{aligned}x_A &= 0x_B + 1x_A + \lambda \\x_B &= 0x_C + 1x_A + \lambda \\x_C &= 0x_C + 1x_C + \emptyset \\ \text{Solución: } x_A &= (01 + 1)^*(0 + \lambda)\end{aligned}$$

3.



Solución: $x_0 = b + c.c^*.a$

4. Autómata no determinista



Solución:

$$x_0 = (b + a.c^*.a)^*.a.c^*$$

7.2.2 AFND equivalente a una Expresión Regular

Dos métodos principales para convertir expresiones regulares en autómatas:

- Método de las derivadas (Alfonseca) \Rightarrow se obtiene una gramática regular que se puede convertir en AFND
- Método de composición de autómatas (Alfonseca, Linz, Hopcroft)

Teorema 2:

Dada una expresión regular R sobre el alfabeto Σ , existe un autómata finito no determinista A tal que $L(R)=L(A)$.

Demostración:

Basándose en la estructura de la expresión regular R , la demostración procede por inducción estructural. Sea $\Sigma=\{a_1,\dots,a_n\}$.

Si R es una expresión regular primitiva:

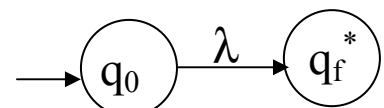
- $R=\emptyset$

A	a_1	\dots	a_n	λ
$\rightarrow q_0$				
$* q_f$				



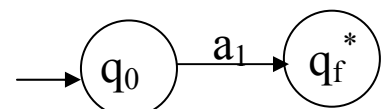
- $R=\{\lambda\}$

A	a_1	\dots	a_n	λ
$\rightarrow q_0$				$\{q_f\}$
$* q_f$				



- $R=\{a_1\}$

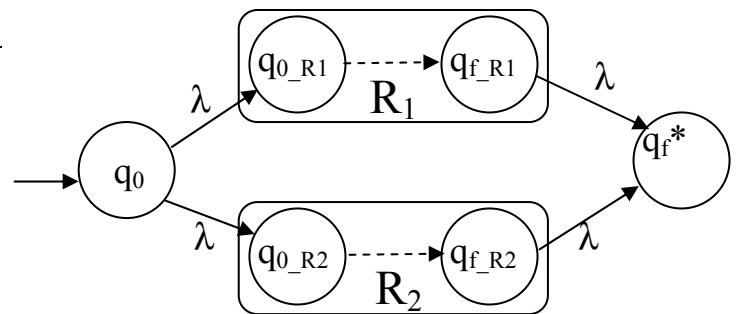
A	a_1	\dots	a_n	λ
$\rightarrow q_0$	$\{q_f\}$			
$* q_f$				



Si R es una expresión regular derivada:

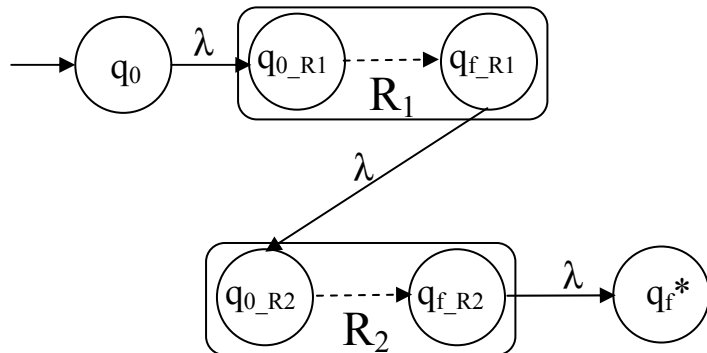
- Si $R=R_1+R_2$:

A	a_1	...	a_n	λ
$\rightarrow q_0$				$\{q_{0_R1}, q_{0_R2}\}$
q_{0_R1}
q_{f_R1}				$\{q_f\}$
q_{0_R2}
q_{f_R2}				$\{q_f\}$
* q_f				



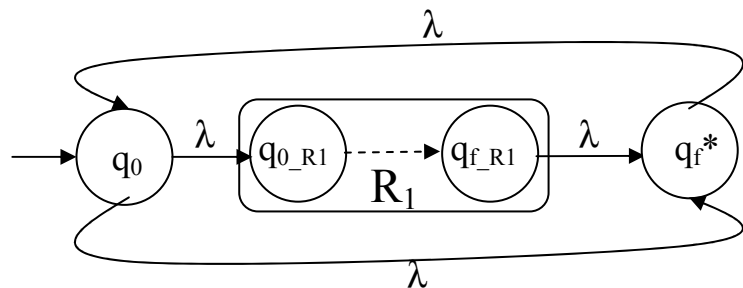
- Si $R=R_1.R_2$:

A	a_1	...	a_n	λ
$\rightarrow q_0$				$\{q_{0_R1}\}$
q_{0_R1}
q_{f_R1}				$\{q_{0_R2}\}$
q_{0_R2}
q_{f_R2}				$\{q_f\}$
* q_f				



- Si $R=R_1^*$:

A	a_1	...	a_n	λ
$\rightarrow q_0$				$\{q_{0_R1}, q_f\}$
q_{0_R1}
q_{f_R1}				$\{q_f\}$
* q_f				$\{q_0\}$

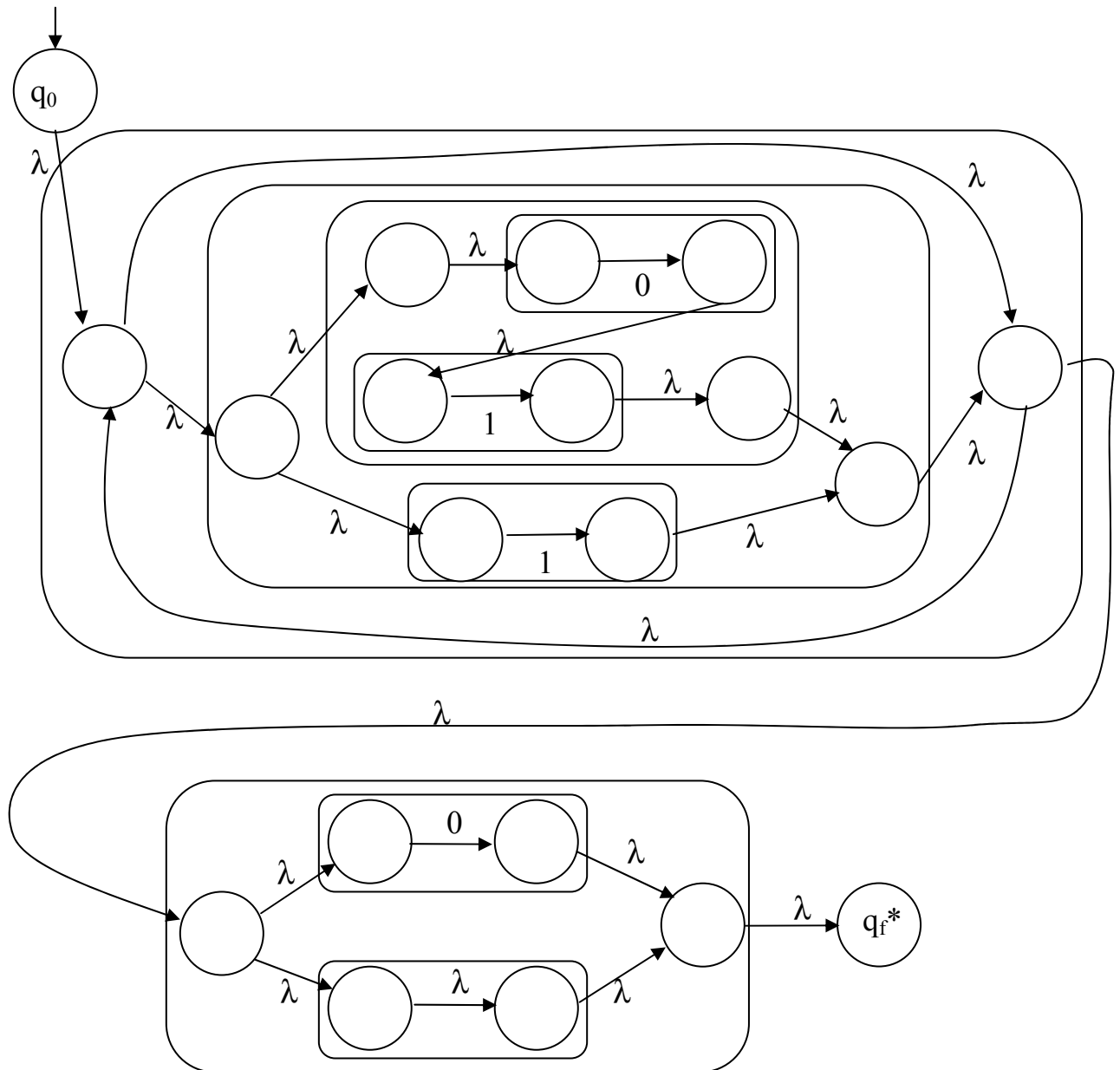


Se construye el autómata de forma recursiva:

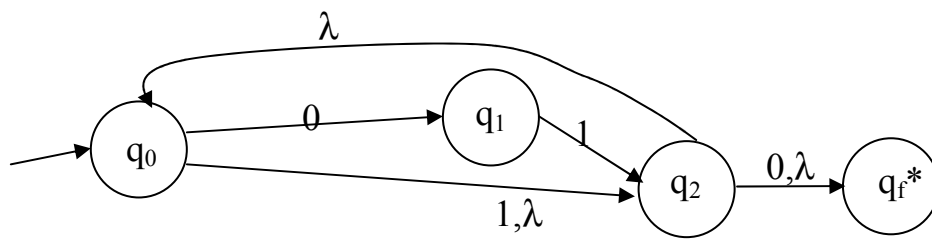
- q_{0_R1} y q_{0_R2} : estados iniciales de los subautómatas para R_1 y R_2 (¡no se marcan como estados iniciales!)
- q_{f_R1} y q_{f_R2} : estados finales de los subautómatas para R_1 y R_2 (¡no se marcarán como estados finales!)

Ejemplos: Obtener los AFND correspondientes a las siguientes expresiones regulares:

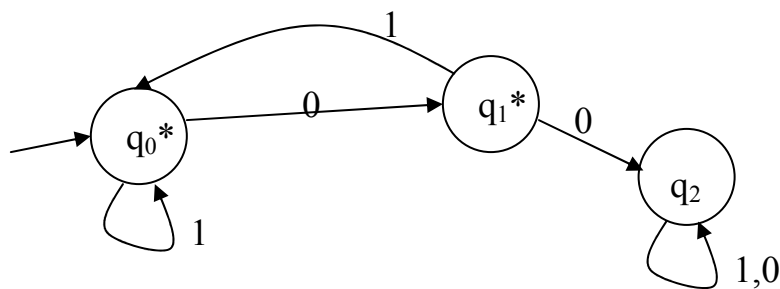
1. $R = (1+01)^*(0+\lambda)$:



Como se puede observar, los autómatas así construidas tienen muchas transiciones que se pueden unir:



El AFD mínimo correspondiente es el siguiente:



2. $R = (1 + 01^*)^*$

3. $R = a.a^*.b.b^*$

4. $R = (b+a).a^*$

Corolario:

Sean L_{REG} , L_{AF} y L_{ER} las clases de los lenguajes aceptados por autómatas finitos, generados por gramáticas regulares y descritos por expresiones regulares, respectivamente.

$$L_{\text{REG}} = L_{\text{AF}} = L_{\text{ER}}$$

