# Tema 5: Expresiones Regulares

U.D. Computación

DSIC - UPV

2017-18

# Índice

# Definiciones



- Inductivamente, una expresión regular sobre $\Sigma$ se define:
  - $\emptyset$ denota el lenguaje vacio
  - $\lambda$ denota el lenguaje $\{\lambda\}$
  - $\forall a \in \Sigma$, $a$ denota el lenguaje $\{a\}$
  - Si $r$ y $s$ son expresiones regulares que denotan $L_r$ y $L_s$:
    - $(r)$ denota el lenguaje $L_r$
    - $r + s$ denota el lenguaje $L_r \cup L_s$
    - $rs$ denota el lenguaje $L_r L_s$
    - $(r)^*$ denota el lenguaje $L_r^*$
  - Sólo son expresiones regulares las construidas de esta forma

# Propiedades

- Sean $\alpha$, $\beta$ y $\gamma$ expresiones regulares
  1. $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$
  2. $\alpha(\beta\gamma) = (\alpha\beta)\gamma$
  3. $\alpha + \beta = \beta + \alpha$
  4. $\alpha(\beta + \gamma) = (\alpha\beta) + (\alpha\gamma)$
  5. $(\alpha + \beta)\gamma = (\alpha\gamma) + (\beta\gamma)$
  6. $\alpha\lambda = \lambda\alpha = \alpha$
  7. $\alpha + \emptyset = \emptyset + \alpha = \alpha$
  8. $\alpha\emptyset = \emptyset\alpha = \emptyset$
  9. $\lambda^* = \lambda$
  10. $\emptyset^* = \lambda$
  11. $\alpha^* = \lambda + \alpha\alpha^*$
  12. $(\alpha^* + \beta^*)^* = (\alpha^*\beta^*)^* = (\alpha + \beta)^*$
  13. $(\alpha\beta)^*\alpha = \alpha(\beta\alpha)^*$
  14. $(\alpha^*\beta)^*\alpha^* = (\alpha + \beta)^*$
  15. $(\alpha^*\beta)^* = (\alpha + \beta)^*\beta + \lambda$

# Construcciones



- Homomorfismo
- Reverso

# Construcciones



- Homomorfismo

  Dada una expresión regular $\alpha$ y un homomorfismo $h : \Sigma_\alpha \to \Delta^*$, para obtener una expresión regular para $h(L(\alpha))$, basta sustituir cada símbolo $a$ de $\alpha$ por $h(a)$

  Por ejemplo, considerando $\alpha = a(bb^* + (aa)^*)^* b$ y el homomorfismo: $h(a) = 0$ y $h(b) = 11$, la expresión regular para $h(L(\alpha))$ sería:

$$0(11(11)^* + (00)^*)^* 11$$

# Construcciones

- Reverso
  Dada una expresión regular $\alpha$, para obtener una expresión regular $\alpha^r$ tal que $L(\alpha^r) = (L(\alpha))^r$, aplicamos recursivamente las siguientes reglas:
  - Si $\alpha = \emptyset$, $\alpha = \lambda$ o $\alpha = a \in \Sigma$, entonces $\alpha^r = \alpha$
  - Si $\alpha = \beta + \gamma$, entonces $\alpha^r = \beta^r + \gamma^r$
  - Si $\alpha = \beta\gamma$, entonces $\alpha^r = \gamma^r\beta^r$
  - Si $\alpha = \beta^*$, entonces $\alpha^r = (\beta^r)^*$

  Por ejemplo, considerando $\alpha = a(b(a+b)^* + (bba)^*)^* b$, la expresión regular para $(L(\alpha))^r$ sería:

$$\alpha^r = b((a+b)^* b + (abb)^*)^* a$$

# Síntesis de AFs a partir de ERs



- Algoritmo de Brzozowski
- Autómata de Posición
- Autómata Follow

# Cálculo de Derivadas



- Reglas para el cálculo de las derivadas
  - Respecto a símbolos ($a, b \in \Sigma, \quad r, s$ E.R.)
    1. $a^{-1}\emptyset = \emptyset$
    2. $a^{-1}\lambda = \emptyset$
    3. $a^{-1}b = \begin{cases} \emptyset & \text{si } a \neq b \\ \lambda & \text{si } a = b \end{cases}$
    4. $a^{-1}(r + s) = a^{-1}r + a^{-1}s$
    5. $a^{-1}(rs) = \begin{cases} (a^{-1}r)s & \text{si } \lambda \notin r \\ (a^{-1}r)s + a^{-1}s & \text{si } \lambda \in r \end{cases}$
    6. $a^{-1}r^* = (a^{-1}r)r^*$
  - Respecto a cadenas ($a \in \Sigma, \quad x \in \Sigma^*$)
    1. $\lambda^{-1}r = r$
    2. $(xa)^{-1}r = a^{-1}(x^{-1}r)$

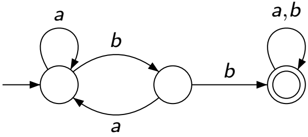
# Algoritmo de Brzozowski

**Entrada:** $\alpha$ expresión regular sobre $\Sigma$
**Salida:** AFD mínimo para $L(\alpha)$
**Metodo:**

$Q = \{\alpha\}$; $q_0 = \alpha$; $F = \emptyset$; $\delta = \emptyset$;
**if** $\lambda \in L(\alpha)$ **then**
    $F = F \cup \{\alpha\}$
**end if**
$activos = \{\alpha\}$
**while** $activos \neq \{\}$ **do**
    $\beta = First(activos)$
    $activos = Rest(activos)$
    **for all** $a \in \Sigma$ **do**
        $\beta' = a^{-1}\beta$
        **if** $\nexists r \in Q \ : \ L(r) = L(\beta')$ **then**
            $Q = Q \cup \{\beta'\}$
            $\delta = \delta \cup \{(\beta, a, \beta')\}$
            $activos = activos \cup \{\beta'\}$
            **if** $\lambda \in L(\beta')$ **then**
                $F = F \cup \{\beta'\}$
            **end if**
        **end if**
    **end for**
**end while**
Return $(Q, \Sigma, \delta, q_0, F)$
**Fin Metodo**

# Algoritmo de Brzozowski. Ejemplo

- Consideremos $\alpha = (a+b)^*bb(a+b)^*$:
  - $q_0 = \alpha = (a+b)^*bb(a+b)^*$; $\lambda \notin L(q_0)$ por lo tanto $F = \emptyset$
  - $a^{-1}q_0 = q_0$
    $b^{-1}q_0 = (a+b)^*bb(a+b)^* + b(a+b)^* = q_1$; $\lambda \notin L(q_1)$ por lo tanto $F = \emptyset$.
  - $a^{-1}q_1 = q_0$
    $b^{-1}q_1 = (a+b)^*bb(a+b)^* + b(a+b)^* + (a+b)^* = (a+b)^* = q_2$; $\lambda \in L(q_2)$ por lo tanto $F = \{q_2\}$.
  - $a^{-1}q_2 = b^{-1}q_2 = q_2$



# Autómata de Posición

- Automata local. Lenguaje Local
- Expresión regular linearizada
- AFD para una expresión regular linearizada
- Autómata de posición

# Automata local. Lenguaje Local



- El AFD $A = (Q, \Sigma, \delta, q_0, F)$ es *local* si y solo si para cualquier $a \in \Sigma$ el conjunto $\{\delta(q, a) \ : \ q \in Q\}$ posee a lo sumo un elemento
- Si además no existe ningún arco que alcance $q_0$, el autómata es *local estandar*
- Un lenguaje es local si y solo si es reconocido por un autómata local estandar

# Expresión regular linearizada

- Sea $\alpha$ una expresión regular y sea $n$ el número de símbolos en $\alpha$ excluyendo paréntesis y símbolos de operación. La expresión linearizada de $\alpha$ (denotada por $\overline{\alpha}$) se obtiene colocando un subíndice $j \in \{1, \ldots, n\}$ a cada símbolo de $\alpha$ indicando su posición.
  p.e.: Siendo

$$\alpha = (a + b)(a^* + ba^* + b^*)^*$$

  la versión linearizada es

$$\overline{\alpha} = (a_1 + b_2)(a_3^* + b_4 a_5^* + b_6^*)^*$$

# Expresión regular linearizada

- Si $\Sigma_\alpha$ y $\Sigma_{\overline{\alpha}}$ son los alfabetos de $\alpha$ y $\overline{\alpha}$ respectivamente, y $h : \Sigma_{\overline{\alpha}}^* \to \Sigma_\alpha^*$ es un homomorfismo que borra los subíndices, entonces:

$$h(L(\overline{\alpha})) = L(\alpha)$$

- Por lo tanto, puede obtenerse un autómata finito para $L(\alpha)$ construyendo un autómata para $L(\overline{\alpha})$ y posteriormente eliminando los subíndices de este autómata (*autómata de posición*)

# AFD para una expresión regular linearizada

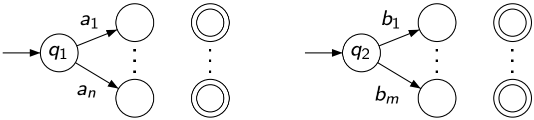


- Toda expresión regular linearizada denota un lenguaje local (reconocido por un AF local estandar)
  Puede verse por inducción sobre la estructura de las expresiones regulares.
- Casos base:


$\emptyset$ $\lambda$ $a$

# AFD para una expresión regular linearizada

- Expresiones compuestas:
  Sean $\overline{\alpha}$ y $\overline{\beta}$ expresiones regulares linearizadas, y sean $A(\overline{\alpha}) = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ y $A(\overline{\beta}) = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$, con $\Sigma_1 \cap \Sigma_2 = \emptyset$, autómatas locales que aceptan $L(\overline{\alpha})$ y $L(\overline{\beta})$ respectivamente:

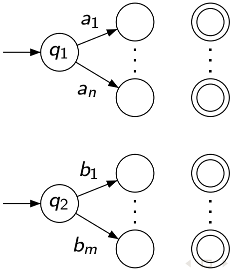



$(\overline{\alpha})$ $(\overline{\beta})$

# AFD para una expresión regular linearizada

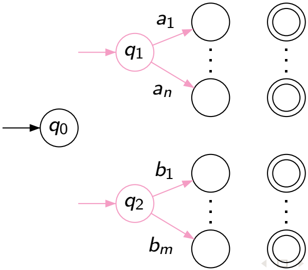


- Unión ($\overline{\alpha} + \overline{\beta}$):
  - $Q = (Q_1 - \{q_1\}) \cup (Q_2 - \{q_2\}) \cup \{q_0\}$, $q_0 \notin Q_1 \cup Q_2$.
  - $\delta = \{(q, a, q') \in \delta_1 \cup \delta_2 : q \notin \{q_1, q_2\}\} \cup \{(q_0, a, q) : (q_1, a, q) \in \delta_1 \vee (q_2, a, q) \in \delta_2\}$,
  - $F = \begin{cases} F_1 \cup F_2 & \text{si } q_1 \notin F_1 \wedge q_2 \notin F_2 \\ (F_1 - \{q_1\}) \cup (F_2 - \{q_2\}) \cup \{q_0\} & \text{en otro caso.} \end{cases}$



# AFD para una expresión regular linearizada

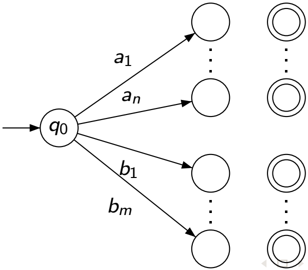
- Unión $(\overline{\alpha}+\overline{\beta})$:
  - $Q = (Q_1 - \{q_1\}) \cup (Q_2 - \{q_2\}) \cup \{q_0\}$, $q_0 \notin Q_1 \cup Q_2$.
  - $\delta = \{(q,a,q') \in \delta_1 \cup \delta_2 : q \notin \{q_1,q_2\}\} \cup \{(q_0,a,q) : (q_1,a,q) \in \delta_1 \vee (q_2,a,q) \in \delta_2\}$,
  - $F = \begin{cases} F_1 \cup F_2 & \text{si } q_1 \notin F_1 \wedge q_2 \notin F_2 \\ (F_1 - \{q_1\}) \cup (F_2 - \{q_2\}) \cup \{q_0\} & \text{en otro caso.} \end{cases}$



# AFD para una expresión regular linearizada

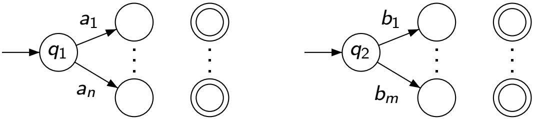


- Unión $(\overline{\alpha} + \overline{\beta})$:
  - $Q = (Q_1 - \{q_1\}) \cup (Q_2 - \{q_2\}) \cup \{q_0\}$, $q_0 \notin Q_1 \cup Q_2$.
  - $\delta = \{(q, a, q') \in \delta_1 \cup \delta_2 : q \notin \{q_1, q_2\}\} \cup \{(q_0, a, q) : (q_1, a, q) \in \delta_1 \vee (q_2, a, q) \in \delta_2\}$,
  - $F = \begin{cases} F_1 \cup F_2 & \text{si } q_1 \notin F_1 \wedge q_2 \notin F_2 \\ (F_1 - \{q_1\}) \cup (F_2 - \{q_2\}) \cup \{q_0\} & \text{en otro caso.} \end{cases}$



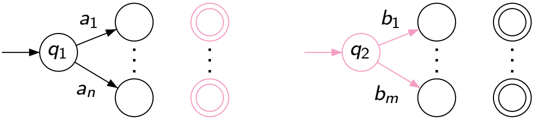
# AFD para una expresión regular linearizada

- Producto $(\overline{\alpha} \cdot \overline{\beta})$ $(q_2 \notin F_2)$:
  - $Q = (Q_1 \cup Q_2) - \{q_2\})$,
  - $\delta = \delta_1 \cup \{(q, a, q') \in \delta_2 : q \neq q_2\} \cup \{(q, a, q') : q \in F_1 \wedge (q_2, a, q') \in \delta_2\}$,
  - $q_0 = q_1$
  - $F = F_2$



# AFD para una expresión regular linearizada

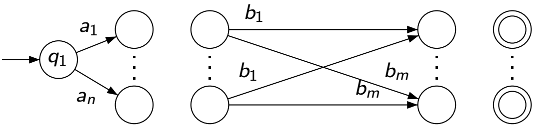
- Producto $(\overline{\alpha} \cdot \overline{\beta})$ $(q_2 \notin F_2)$:
  - $Q = (Q_1 \cup Q_2) - \{q_2\})$,
  - $\delta = \delta_1 \cup \{(q, a, q') \in \delta_2 : q \neq q_2\} \cup \{(q, a, q') : q \in F_1 \wedge (q_2, a, q') \in \delta_2\}$,
  - $q_0 = q_1$
  - $F = F_2$



# AFD para una expresión regular linearizada



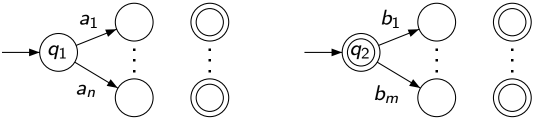
- Producto $(\overline{\alpha} \cdot \overline{\beta})$ $(q_2 \notin F_2)$:
  - $Q = (Q_1 \cup Q_2) - \{q_2\})$,
  - $\delta = \delta_1 \cup \{(q, a, q') \in \delta_2 : q \neq q_2\} \cup \{(q, a, q') : q \in F_1 \wedge (q_2, a, q') \in \delta_2\}$,
  - $q_0 = q_1$
  - $F = F_2$



# AFD para una expresión regular linearizada

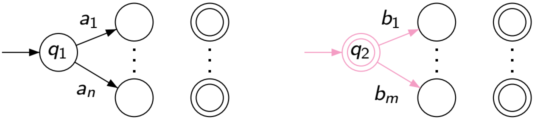


- Producto $(\overline{\alpha} \cdot \overline{\beta})$ $(q_2 \in F_2)$:
  - $Q = (Q_1 \cup Q_2) - \{q_2\}),$
  - $\delta = \delta_1 \cup \{(q, a, q') \in \delta_2 : q \neq q_2\} \cup \{(q, a, q') : q \in F_1 \wedge (q_2, a, q') \in \delta_2\},$
  - $q_0 = q_1$
  - $F = F_1 \cup (F_2 - \{q_2\})$



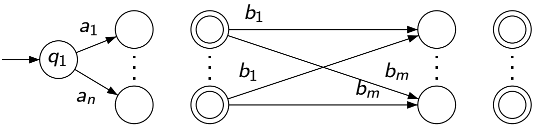
# AFD para una expresión regular linearizada

- Producto $(\overline{\alpha} \cdot \overline{\beta})$ $(q_2 \in F_2)$:
  - $Q = (Q_1 \cup Q_2) - \{q_2\})$,
  - $\delta = \delta_1 \cup \{(q, a, q') \in \delta_2 : q \neq q_2\} \cup \{(q, a, q') : q \in F_1 \wedge (q_2, a, q') \in \delta_2\}$,
  - $q_0 = q_1$
  - $F = F_1 \cup (F_2 - \{q_2\})$



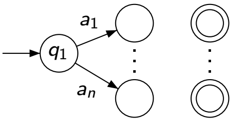
# AFD para una expresión regular linearizada

- Producto $(\overline{\alpha} \cdot \overline{\beta})$ $(q_2 \in F_2)$:
  - $Q = (Q_1 \cup Q_2) - \{q_2\})$,
  - $\delta = \delta_1 \cup \{(q, a, q') \in \delta_2 : q \neq q_2\} \cup \{(q, a, q') : q \in F_1 \wedge (q_2, a, q') \in \delta_2\}$,
  - $q_0 = q_1$
  - $F = F_1 \cup (F_2 - \{q_2\})$



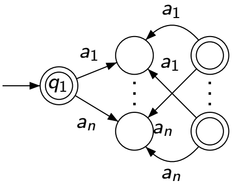
# AFD para una expresión regular linearizada

- Clausura ($\overline{\alpha}^*$):
  - $\delta' = \delta \cup \{(q, a, q') : q \in F \wedge (q_0, a, q') \in \delta\}$
  - $F = F_1 \cup \{q_1\})$



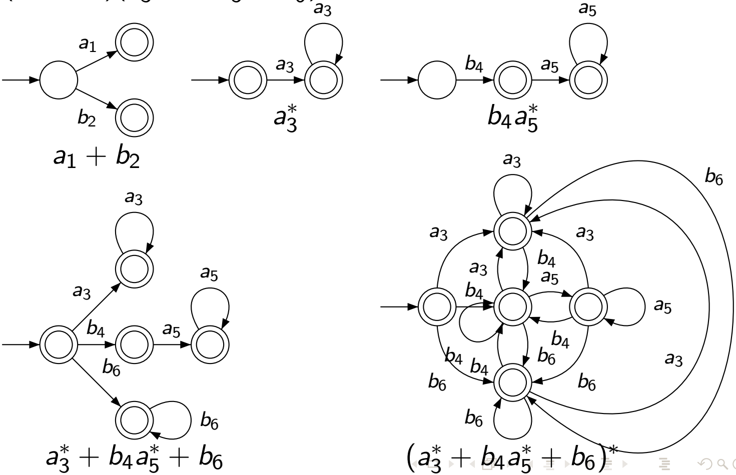
# AFD para una expresión regular linearizada

- Clausura ($\overline{\alpha}^*$):
  - $\delta' = \delta \cup \{(q, a, q') : q \in F \wedge (q_0, a, q') \in \delta\}$
  - $F = F_1 \cup \{q_1\})$



# AFD para una expresión regular linearizada. Ejemplo

Sea $\alpha = (a + b)(a^* + ba^* + b^*)^*$. Entonces
$\overline{\alpha} = (a_1 + b_2)(a_3^* + b_4 a_5^* + b_6^*)^*$



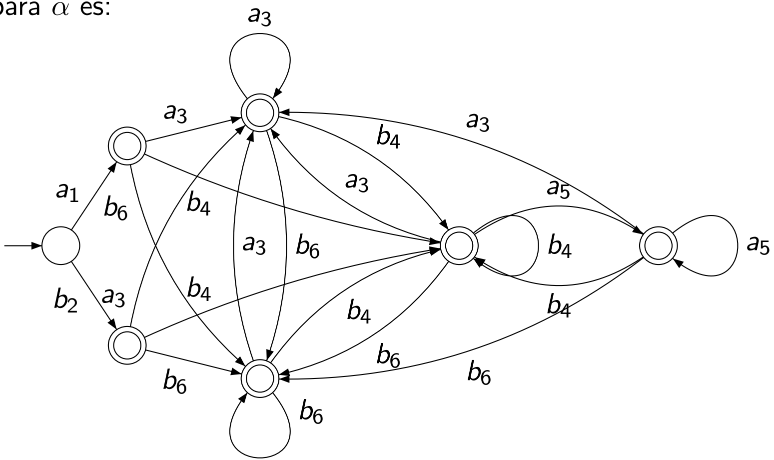
# Autómata de posición. Algoritmo



**Entrada:** $\alpha$ expresión regular sobre $\Sigma$
**Salida:** AF para $L(\alpha)$
**Método:**
Obtener $\overline{\alpha}$ versión linearizada de $\alpha$
Obtener $A$ un Autómata local estandar para $\overline{\alpha}$
$A_{pos} = h(A)$, donde $h$ es un homomorfismo de borrado de los subíndices.
Return $A_{pos}$
**Fin Método**

# Autómata de posición. Ejemplo

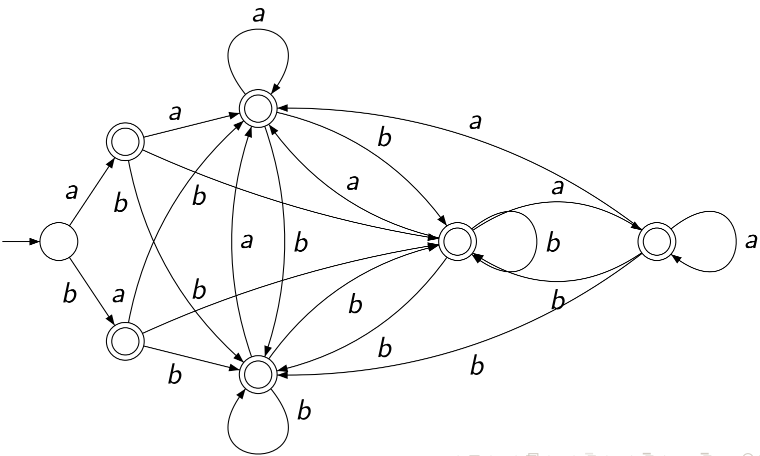
Dada $\alpha = (a + b)(a^* + ba^* + b^*)^*$ y su versión linearizada $\overline{\alpha} = (a_1 + b_2)(a_3^* + b_4 a_5^* + b_6^*)^*$, el autómata local estandar para $\overline{\alpha}$ es:



# Autómata de Posición. Ejemplo



y el autómata de posición para $\alpha = (a + b)(a^* + ba^* + b^*)^*$ es:



# Autómata Follow

- Relación follow
- Autómata follow

# Autómata Follow



- El *autómata follow* de una expresión regular $\alpha$ se propone como el automata cociente del autómata de posición por la siguiente relación:

$$p \equiv_f q \Leftrightarrow \begin{cases} p, q \in F \ o \ bien \ p, q \in Q - F \\ follow(p) = follow(q) \end{cases}$$

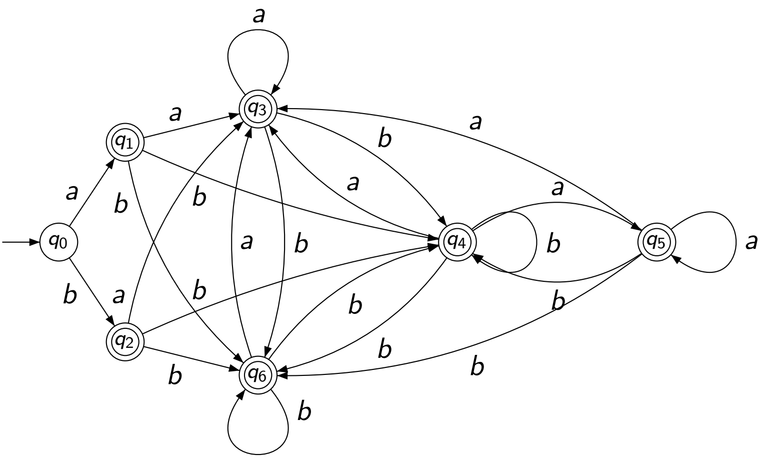
donde $follow(p) = \{q \in Q \ : \ \exists a \in \Sigma, \delta(p, a) = q\}$

- El autómata cociente resultante es una reducción parcial del autómata de posición.

# Autómata Follow

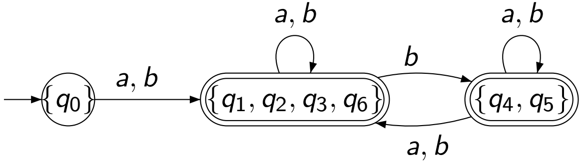


Recordamos el autómata de posición para $\alpha = (a+b)(a^* + ba^* + b^*)^*$:



# Autómata Follow

Las clases de equivalencia son: $\{q_0\}, \{q_1, q_2, q_3, q_6\}, \{q_4, q_5\}$, con lo que el autómata follow para $\alpha$ queda:



# Análisis de autómatas finitos

- Sistemas de ecuaciones en expresiones regulares
- Lema de Arden
- Análisis de autómatas finitos

# Sistemas de ecuaciones en expresiones regulares. Lema de Arden

- Ecuación en expresiones regulares: Ecuación lineal donde variables y coeficientes toman la forma de expresiones regulares.

$$X = rX + s$$

- Lema de Arden: Sea $X = rX + s$ una ecuación en expresiones regulares. $X = r^*s$ es una solución para la ecuación. Es única si $\lambda \notin r$
  - demostramos que $r^*s$ es solución:

$$rX + s \underset{X=r^*s}{=} rr^*s + s = (rr^* + \lambda)s \underset{rr^*+\lambda=r^*}{=} r^*s$$

  - Si $\lambda \in r$ existen infinitas soluciones: $\forall t \subseteq \Sigma^*$, $r^*(s+t)$ es solución:

$$\begin{aligned} X =& rX + s = rr^*(s+t) + s = rr^*s + rr^*t + s = \\ =& (rr^* + \lambda)s + rr^*t \underset{rr^*+\lambda=r^*}{=} r^*s + r^*t \underset{X=r^*(s+t)}{=} X \end{aligned}$$

# Sistemas de ecuaciones en expresiones regulares

- Dado un sistema de ecuaciones en expresiones regulares:

$$\left\{\begin{array}{ll} X_1 = r_{11}X_1 + r_{12}X_2 + & \ldots + r_{1n}X_n + s_1 \\ X_2 = r_{21}X_1 + r_{22}X_2 + & \ldots + r_{2n}X_n + s_2 \\ & \ldots \\ X_n = r_{n1}X_1 + r_{n2}X_2 + & \ldots + r_{nn}X_n + s_n \end{array}\right.$$

  la resolución viene tras aplicar el método de Gauss utilizando el Lemma de Arden para reducir.

# Análisis de AFs. Algoritmo

**Entrada:** Autómata finito $A = (Q, \Sigma, \delta, q_1, F)$ con $Q = \{q_1, q_2, \ldots, q_n\}$
**Salida:** Expresión regular para $L(A)$
**Metodo:**
Por cada estado $q_i$ introducir una variable $X_i$
Si $q_i \in F$ entonces en la parte derecha de la $i$-esima ecuación aparece el término $\lambda$
Si $q_j \in \delta(q_i, a)$ entonces en la parte derecha de la $i$-esima ecuación aparece el término $aX_j$, con $a \in \Sigma \cup \{\lambda\}$
Resolver el sistema de ecuaciones en expresiones regulares utilizando el Lema de Arden para reducir
Devolver la expresión regular asociada al estado inicial
**Fin Metodo**