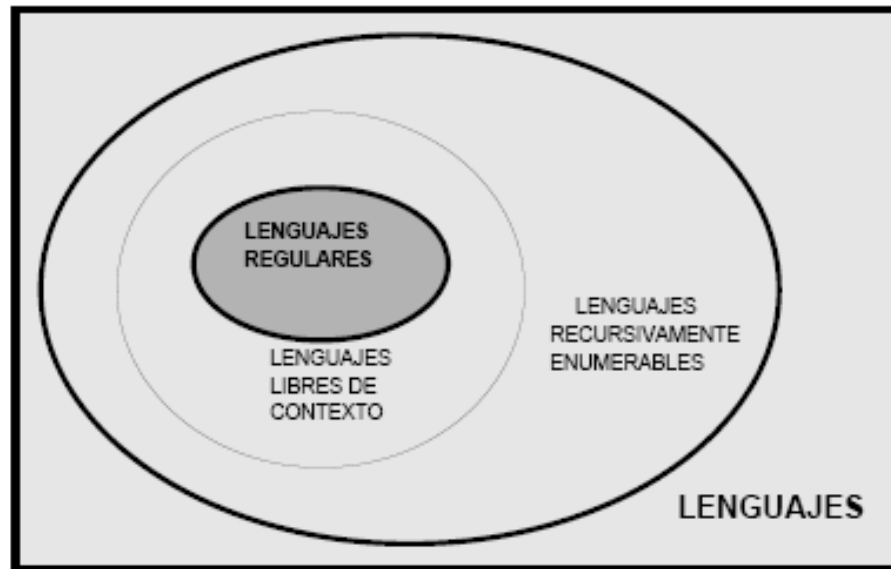


Expresiones regulares, gramáticas regulares

Unidad 3

Los LR en la jerarquía de Chomsky

- La clasificación de lenguajes en clases de lenguajes se debe a N. Chomsky, quien propuso una jerarquía de lenguajes, donde las clases más complejas incluyen a las más simples.



-
- Los “lenguajes regulares” es la clase más pequeña, e incluye a los lenguajes más simples. Por ejemplo, el conjunto de todos los números binarios.
 - Los “lenguajes libres de contexto” incluyen a los LR. Por ejemplo, la mayoría de los lenguajes de programación.
 - Los “lenguajes recursivamente enumerables” que incluyen a los dos anteriores.
-

Lenguajes regulares

- Los LR se llaman así porque sus palabras contienen “regularidades” o repeticiones de los mismos componentes, por ejemplo:
 - $L1 = \{ab, abab, ababab, abababab, \dots\}$
- Las palabras de $L1$ son simplemente repeticiones de “ab” cualquier número de veces. La “regularidad” consiste en que las palabras contienen “ab” algún número de veces.
- Otro ejemplo:
 - $L2 = \{abc, cc, abab, abccc, ababc, \dots\}$
 - La regularidad consiste en que sus palabras inician con repeticiones de “ab” seguidas de repeticiones de “c”.
- Los lenguajes finitos son también regulares por definición.
Ejemplo: $L3 = \{el, coche, verde\}$
- La combinación de lenguajes regulares (unión o concatenación), también producen un lenguaje regular.

Definición de Lenguajes Regulares

- Un lenguaje L es regular si y sólo si se cumple al menos una de las siguientes condiciones:
 - L es finito,
 - L es la unión o la concatenación de otros lenguajes regulares R_1 y R_2 , $L=R_1 \cup R_2$ o $L=R_1 R_2$ respectivamente.
 - L es la cerradura de Kleene de algún lenguaje regular, $L=R^*$.

Ejemplo

- Sea el lenguaje L de palabras formadas por a y b pero que empiezan con a , como aab , ab , a , $abaa$, etc. Probar que este lenguaje es regular, y dar una expresión de conjuntos que lo represente.
 - El alfabeto es $\Sigma = \{a, b\}$. El lenguaje L puede ser visto como la concatenación de una a con cadenas cualesquiera de a y b ; ahora bien, éstas últimas son los elementos de $\{a, b\}^*$, mientras que el lenguaje que sólo contiene la palabra a es $\{a\}$. Ambos lenguajes son regulares. Es decir, $\{a\}$ es finito, por lo tanto regular, mientras que $\{a, b\}^*$ es la cerradura de $\{a, b\}$, que es regular por ser finito. Entonces la concatenación es $\{a\}\{a, b\}^*$, es regular.

- Sea Σ un alfabeto. La expresión regular sobre Σ y los conjuntos que ellas denotan son definidos recursivamente como sigue:
 - \emptyset es una expresión regular y denota el conjunto vacío.
 - ε es una expresión regular y denota el conjunto $\{\varepsilon\}$.
 - Para cada a en Σ , a es una expresión regular y denota el conjunto $\{a\}$.
 - Si r y s son expresiones regulares denotando el lenguaje R y S , respectivamente, entonces $(r + s)$, (rs) , y (r^*) son expresiones regulares que denotan los conjuntos $R \cup S$, RS , y R^* , respectivamente.

Ejemplo

- Sea $L1=\{10,1\}$ y $L2=\{011,11\}$. Entonces
 - $L1L2=\{10011, 1011, 111\}$
 - $\{10, 11\}^* = \{ \epsilon, 10, 11, 1010, 1011, 1110, 1111, \dots \}$
- Observación: L^+ contiene a ϵ sí y sólo si L lo tiene.
- En la escritura de expresiones regulares, se pueden omitir paréntesis si asumimos que $*$ tiene más alta precedencia que la concatenación o $+$, y que la concatenación tiene más alta precedencia que $+$

Definición

- Sea Σ un alfabeto. Las expresiones regulares (ER) definidas sobre Σ y los conjuntos regulares que denotan se definen recursivamente:

| <i>ER</i> | <i>Conjunto</i> |
|-------------|-----------------|
| \emptyset | \emptyset |
| a | $\{a\}$ |
| ϵ | $\{\epsilon\}$ |
| $(r + s)$ | $R \cup S$ |
| (rs) | RS |
| (r^*) | R^* |

- Donde r, s son ERs que denotan a los conjuntos R, S respectivamente. Nótese que los paréntesis sólo son agrupadores.

Ejemplo

- Sea la ER $(0 + 1)^*$. Analizando detalladamente (obteniendo los correspondientes conjuntos para cada subexpresión):
 - 0 es la ER que denota al conjunto $\{0\}$,
 - 1 es la ER que denota al conjunto $\{1\}$,
 - $(0 + 1) = \{0\} \cup \{1\} = \{0,1\}$
- Entonces:
 - $\{0,1\}^* = \{\epsilon, 0, 1, 01, 10, 11, 00, 000, 001, \dots\}$
 - Representa el conjunto de todas las posibles cadenas que se pueden formar con '0' y '1'.

Jerarquía de prioridad y asociatividad

- La cerradura $*$ asocia por la izquierda y tiene la mayor prioridad, a continuación la concatenación (asocia por la derecha) y finalmente el operador de alternativa $+$ que también asocia por la derecha.
- Ejemplo: $0 + 1b^* + a1^*$ es igual a
 - $(0 + ((1(b^*)) + (a(1^*))))$

Ejemplos

- Sea el alfabeto $\Sigma = \{a, b, c\}$,

1. $(a + b)c^*$:

$$a = \{a\}$$

$$b = \{b\}$$

$$c = \{c\}$$

$$(a + b) = \{a, b\}$$

$$c^* = \{\epsilon, c, cc, ccc, cccc, \dots\}$$

$$(a + b)c^* = \{a, b, ac, bc, acc, bcc, \dots\}$$

- La ER denota al conjunto de todas las cadenas que comienzan con 'a' o 'b' seguidas de cualquier número de 'c's incluyendo ϵ .

Ejemplo

2. $(\mathbf{a^*c}) + (\mathbf{ab^*})$.

Tomamos $r = (\mathbf{a^*c})$ y $s = \mathbf{ab^*}$

$$\begin{aligned} r : \quad & \mathbf{a^*} = \{\epsilon, a, aa, aaa, aaaa, \dots\} \\ & \mathbf{a^*c} = \{c, ac, aac, aaac, aaaac, \dots\} \end{aligned}$$

$$\begin{aligned} s : \quad & \mathbf{b^*} = \{\epsilon, b, bb, bbb, bbbb, \dots\} \\ & \mathbf{ab^*} = \{a, ab, abb, abbb, abbbb, \dots\} \end{aligned}$$

$$r + s = \{a, c, ac, ab, aac, abb, aaac, abbbb, ..\}$$

Ejemplo

3. Sea la ER $(00)^* + (1(11)^*)$ definida sobre el alfabeto $\{0,1\}$. Tomamos $r = (00)^*$ y $s = 1(11)^*$.

$$\begin{aligned} r : \quad & (00) = \{00\} \\ & (00)^* = \{\epsilon, 00, 0000, 000000, \dots\} \end{aligned}$$

$$\begin{aligned} s : \quad & (11) = \{11\} \\ & (11)^* = \{\epsilon, 11, 1111, 111111, \dots\} \\ & 1(11)^* = \{1, 111, 11111, 1111111, \dots\} \end{aligned}$$

$$r + s = \{\epsilon, 00, 0000, \dots, 1, 111, 11111, \dots\}$$

- Son todas las cadenas pares de 0's ó cadenas impares de 1's.

Ejercicio

- Determine el conjunto de “ $((a + b))^*a$ ”

Solución

- $\{a,b\}^*\{a\}$ es el lenguaje sobre $\{a,b\}$ de las palabras que terminan en a.

Ejercicio

- Dada la expresión regular $E=0^*10^*$, obtenga el lenguaje que representa

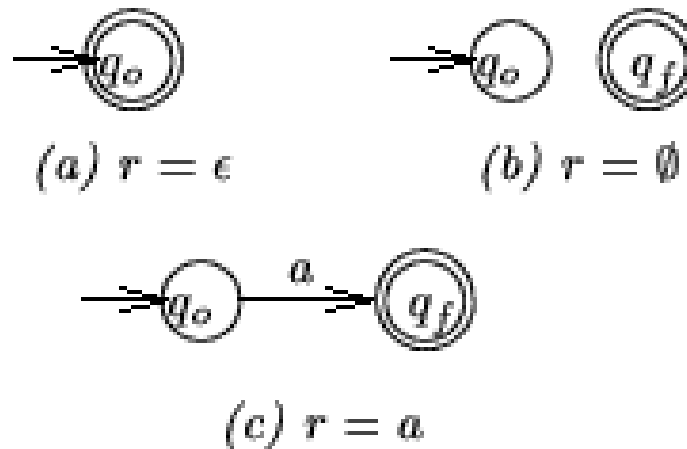
Solución

- $\{0\}^*\{1\}\{0\}^* = \{0^n10^m \mid n, m \geq 0\}$

Equivalencia entre AF y ER

- Teorema. Sea r una expresión regular que denota al conjunto R . Existe un AFND- ϵ M que acepta al lenguaje que denota r : $L(M)=R$.
- Demostración: aplicaremos inducción sobre el número de operadores involucrados en la ER r , el AFND- ϵ tendrá un estado final y ninguna transición fuera de ese estado final, tal que $L(M)=L(r)$.

- Caso base: el número de operadores es 0 (no hay alguno), la ER r corresponde a alguna de \emptyset , ϵ , a como se muestra en la Figura para los casos base.



- Paso inductivo: Se asume que la afirmación es verdadera para toda ER con menos de i operadores, para $i \geq 1$. Sea r con i operadores:

1. $r = r_1 + r_2$, ambos r_1 y r_2 tienen menos de i operadores. Aplicando la HI: existen AFND- ϵ

$$M_1 = \langle Q_1, \Sigma_1, \delta_1, q_{o1}, \{f_1\} \rangle$$

$$M_2 = \langle Q_2, \Sigma_2, \delta_2, q_{o2}, \{f_2\} \rangle$$

tales que $L(M_1) = L(r_1)$ y $L(M_2) = L(r_2)$. Se asume que Q_1 y Q_2 son disjuntos.

El AFND- ϵ resultante será

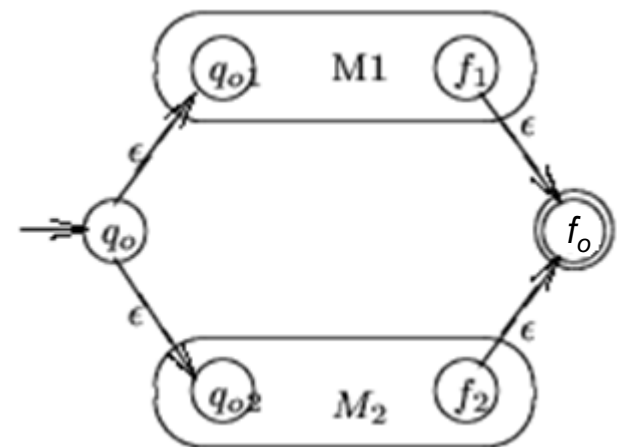
$$M = \langle Q_1 \cup Q_2 \cup \{q_o, f_o\}, \Sigma_1 \cup \Sigma_2, \delta, q_o, \{f_o\} \rangle$$

donde

$$\delta(q_o, \epsilon) = \{q_{o1}, q_{o2}\}$$
$$\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 - \{f_1\}, \\ & a \in \Sigma_1 \cup \{\epsilon\} \\ \delta_2(q, a) & q \in Q_2 - \{f_2\}, \\ & a \in \Sigma_2 \cup \{\epsilon\} \end{cases}$$
$$\delta(f_1, \epsilon) = \{f_o\} = \delta(f_2, \epsilon)$$

Por tanto :

$$L(M) = L(M_1) \cup L(M_2)$$



2. $r = r_1 r_2$. Sean M_1, M_2 como el caso anterior.
Entonces

$$M = \langle Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, q_{o1}, \{f_2\} \rangle$$

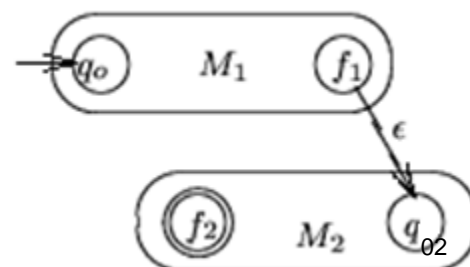
donde

$$\delta(q_o, a) = \begin{cases} \delta_1(q, a) & \text{para } q \in Q_1 - \{f_1\}, \\ & a \in \Sigma_1 \cup \{\epsilon\}. \\ \delta_2(q, a) & \text{para } q \in Q_2, \\ & a \in \Sigma_2 \cup \{\epsilon\}. \end{cases}$$

$$\delta(f_1, \epsilon) = \{q_{o2}\}.$$

Por tanto:

$$L(M) = L(M_1)L(M_2)$$



3. $r = r_1^*$. Sea M_1 como los casos anteriores

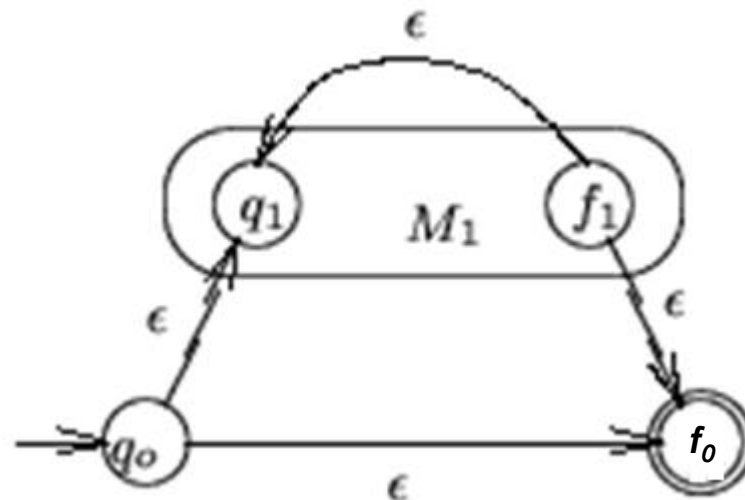
$$M = \langle Q_1 \cup \{q_o, f_o\}, \Sigma_1, \delta_1, q_o, \{f_o\} \rangle$$

donde

$$\hat{\delta}(q_o, \epsilon) = \hat{\delta}_1(f_1, \epsilon) = \{q_1, f_o\}$$

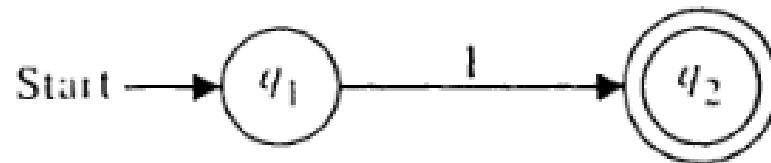
$$\hat{\delta}(q, a) = \delta_1(q, a), \text{ para } q \in Q_1 - \{f_1\}, \\ a \in \Sigma_1 \cup \{\epsilon\}$$

Por lo tanto: $L(M) = L(M_1)^*$

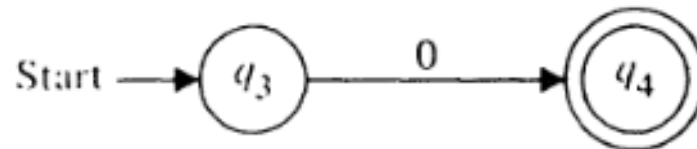


Ejemplo

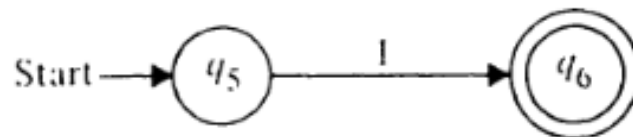
- Construir un AFN-e para la expresión regular $01^* + 1$. Esta expresión es: $(0(1^*)) + 1$, así
 - $r1 = 01^*$ y $r2 = 1$. El autómata para $r2$ es:



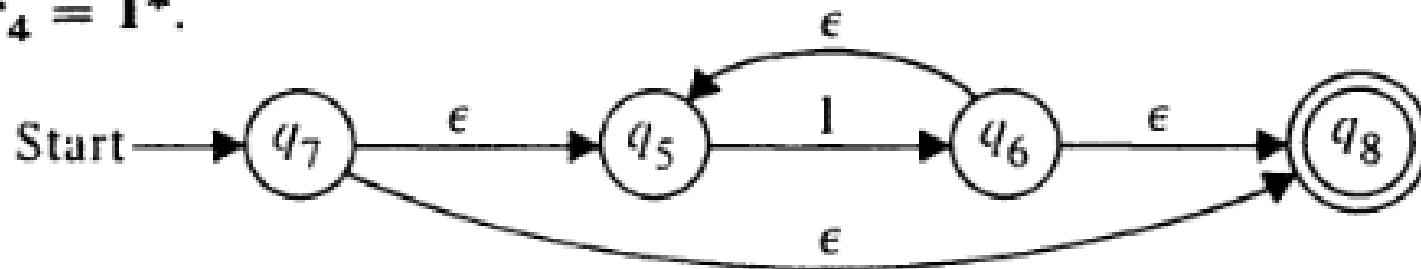
- Ahora, $r1 = r3r4$, donde $r3 = 0$ y $r4 = 1^*$. El autómata para $r3$ es



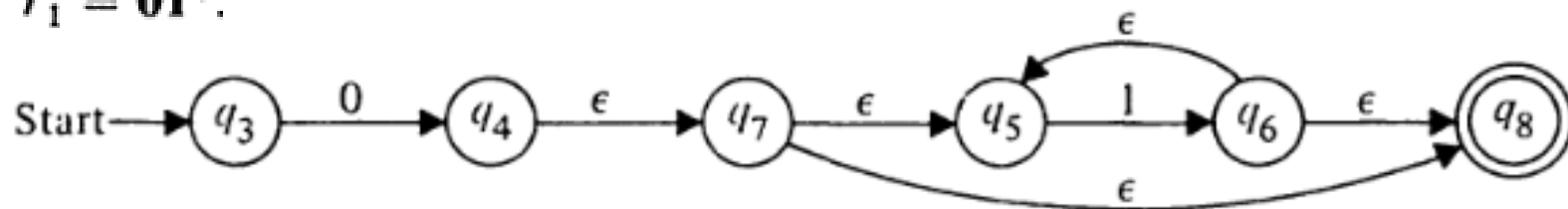
- $R4$ es $r5^*$, donde $r5$ es 1. Un AFN para $r5$ es



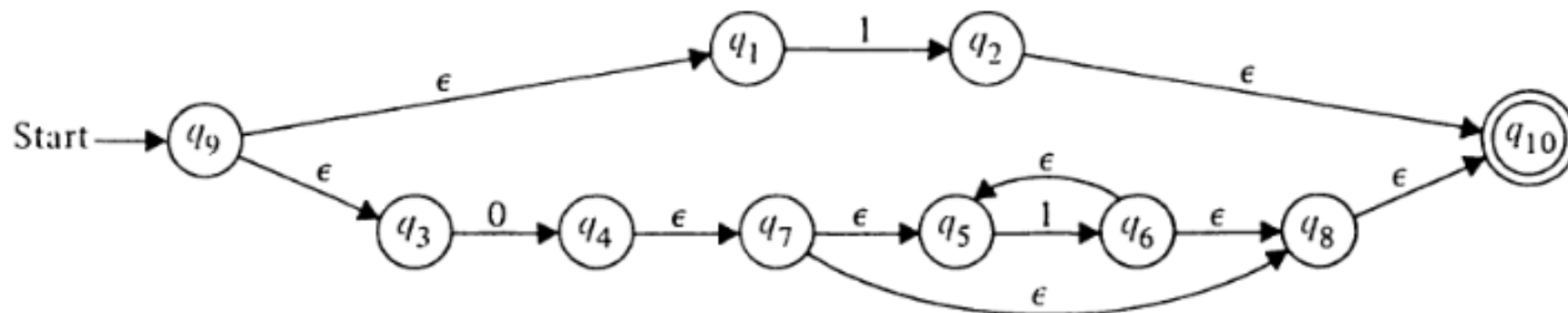
$$r_4 = 1^*$$



$$r_1 = 01^*$$

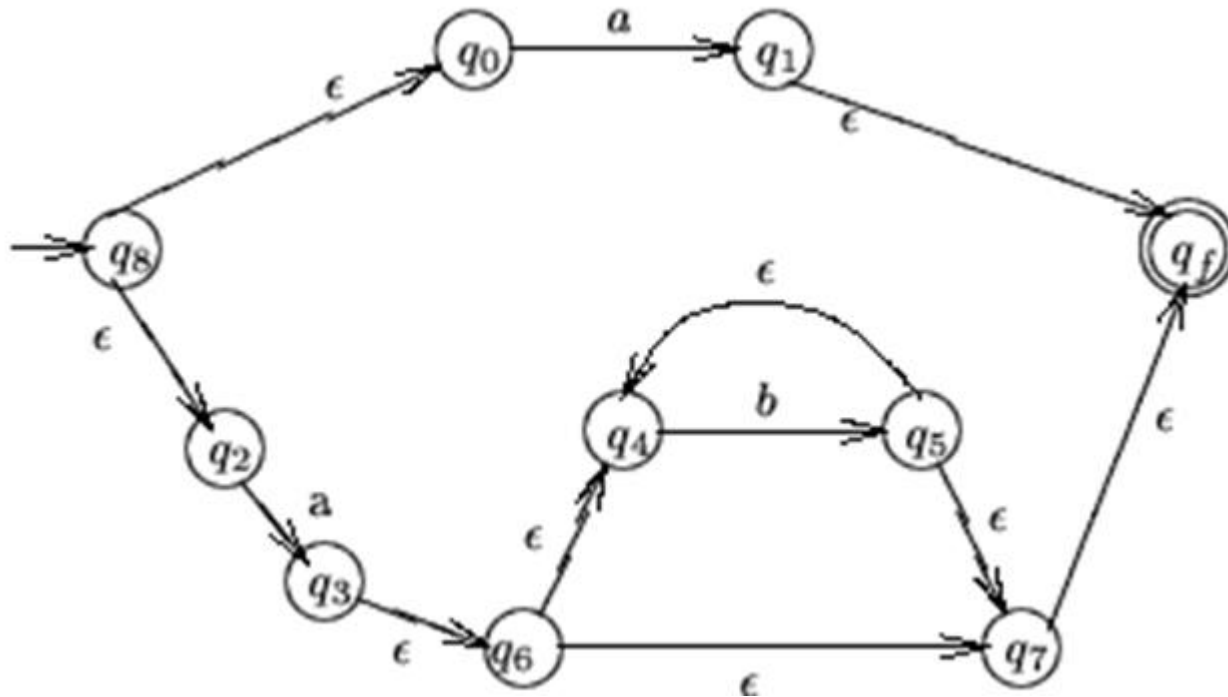


$$r = 01^* + 1$$



Ejemplo

- Construir un AFND- ϵ para la ER ab^*+a .

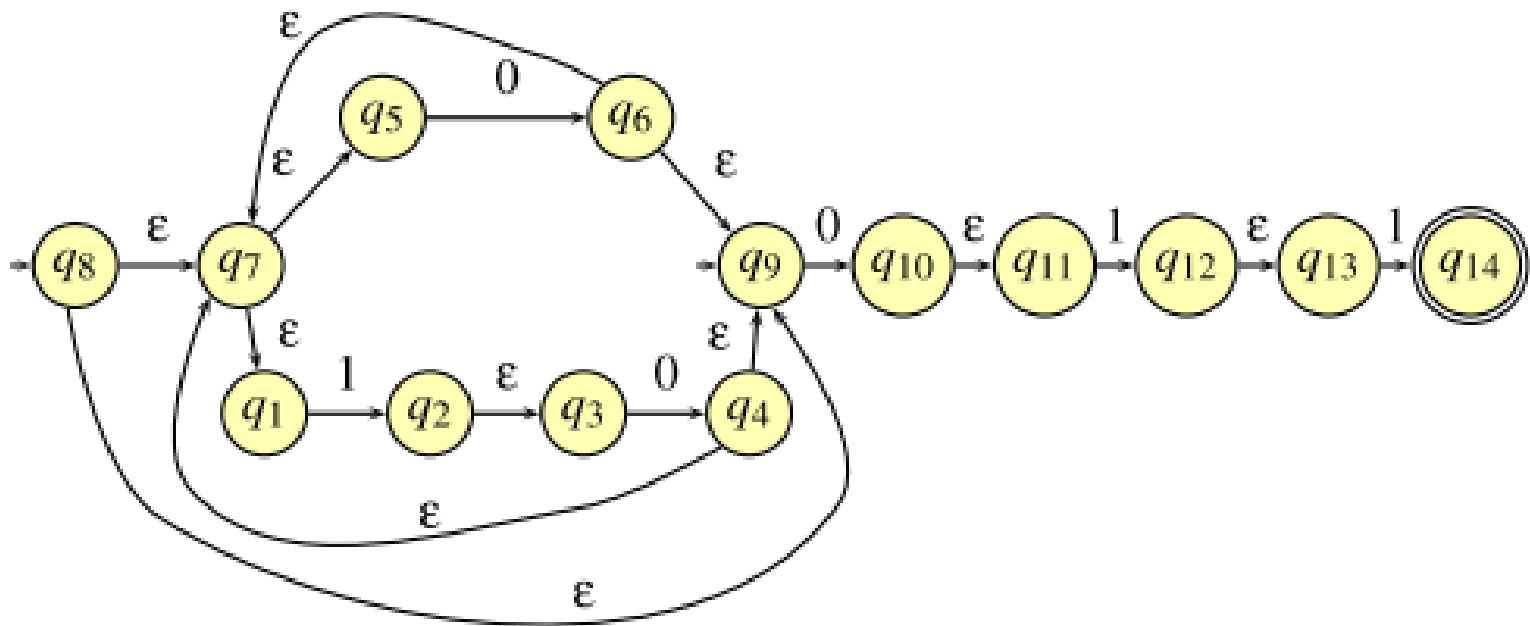


Ejercicio

- Obtener el AF asociado a $(10+0)^*011$

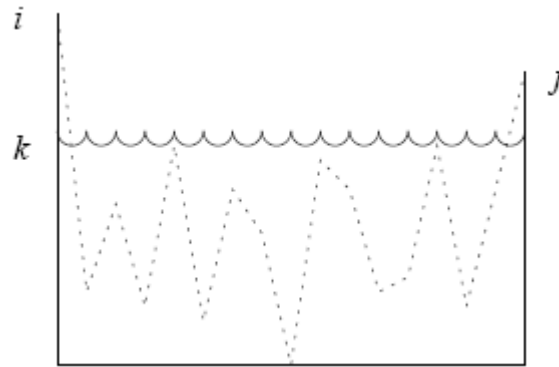
Propuesta de solución

■ $(10+0)^*011$



Teorema:

- Para cada AFD $A=(Q, \Sigma, \delta, q_0, F)$ existe una exp reg R . $L(R) = L(A)$.
- Prueba. Sea los estados de $A=\{1,2,\dots, n\}$ con 1 como el estado inicial.
- Sea $R_{ij}^{(k)}$ una exp reg describiendo el conjunto de etiquetas de todas las rutas en A desde el estado i al estado j que van a través de estados intermedios $\{1,\dots, k\}$ únicamente.



- $R_{ij}^{(k)}$ se define inductivamente. Note que

$$L \left(\bigoplus_{j \in F} R_{1j}^{(n)} \right) = L(A)$$

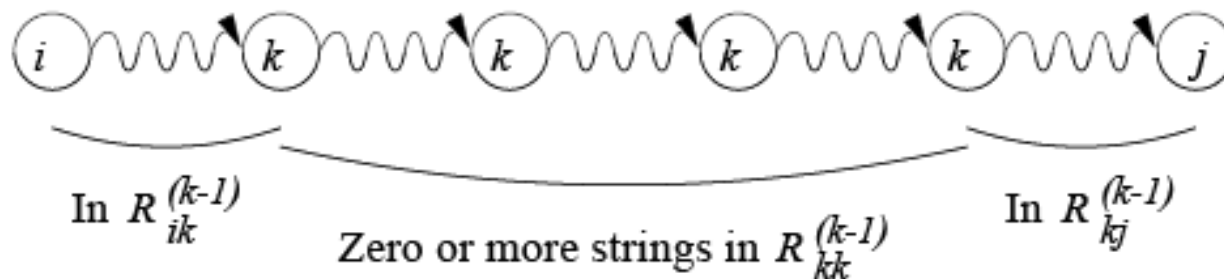
- Base: $k=0$, es decir, no hay estados intermedios.

- Caso 1: $i \neq j$ $R_{ij}^{(0)} = \bigoplus_{\{a \in \Sigma : \delta(i,a)=j\}} a$

- Caso 2: $i=j$ $R_{ii}^{(0)} = \left(\bigoplus_{\{a \in \Sigma : \delta(i,a)=i\}} a \right) + \epsilon$

■ Inductivo

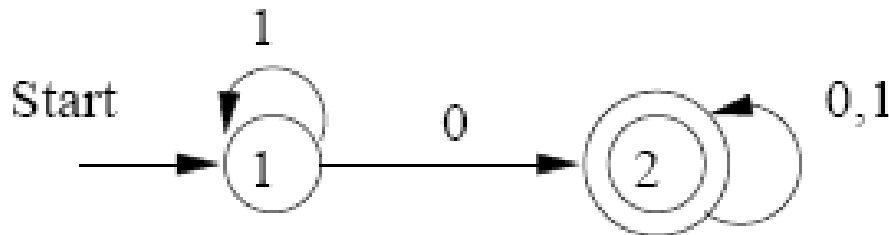
$$\begin{aligned}
 R_{ij}^{(k)} &= \\
 R_{ij}^{(k-1)} &+ \\
 R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)} \right)^* R_{kj}^{(k-1)}
 \end{aligned}$$



Ejemplo

- Obtener la exp reg del siguiente AFD A

$$L(A) = \{x0y : x \in \{1\}^* \text{ and } y \in \{0, 1\}^*\}$$



| $R_{11}^{(0)}$ | $\epsilon + 1$ |
|----------------|--------------------|
| $R_{12}^{(0)}$ | 0 |
| $R_{21}^{(0)}$ | \emptyset |
| $R_{22}^{(0)}$ | $\epsilon + 0 + 1$ |

Reglas de simplificación

- $(\varepsilon + R)^* = R^*$
- $R + RS^* = RS^*$
- $\emptyset R = R\emptyset = \emptyset$ (Aniquilador)
- $\emptyset + R = R + \emptyset = R$ (Identidad)
- $R + SR = (\varepsilon + S)R$

| | |
|----------------|--------------------|
| $R_{11}^{(0)}$ | $\epsilon + 1$ |
| $R_{12}^{(0)}$ | 0 |
| $R_{21}^{(0)}$ | \emptyset |
| $R_{22}^{(0)}$ | $\epsilon + 0 + 1$ |

- $(\epsilon + R)^* = R^*$
- $R + RS^* = RS^*$
- $\emptyset R = R\emptyset = \emptyset$
- $\emptyset + R = R + \emptyset = R$

$$R_{ij}^{(1)} = R_{ij}^{(0)} + R_{i1}^{(0)} \left(R_{11}^{(0)} \right)^* R_{1j}^{(0)}$$

| | | |
|----------------|---|--------------------|
| $R_{11}^{(1)}$ | $\epsilon + 1 + (\epsilon + 1)(\epsilon + 1)^*(\epsilon + 1)$ | 1^* |
| $R_{12}^{(1)}$ | $0 + (\epsilon + 1)(\epsilon + 1)^*0$ | 1^*0 |
| $R_{21}^{(1)}$ | $\emptyset + \emptyset(\epsilon + 1)^*(\epsilon + 1)$ | \emptyset |
| $R_{22}^{(1)}$ | $\epsilon + 0 + 1 + \emptyset(\epsilon + 1)^*0$ | $\epsilon + 0 + 1$ |

| $R_{11}^{(1)}$ | 1^* |
|----------------|--------------------|
| $R_{12}^{(1)}$ | 1^*0 |
| $R_{21}^{(1)}$ | \emptyset |
| $R_{22}^{(1)}$ | $\epsilon + 0 + 1$ |

- $(\epsilon + R)^* = R^*$
- $R + RS^* = RS^*$
- $\emptyset R = R\emptyset = \emptyset$
- $\emptyset + R = R + \emptyset = R$

$$R_{ij}^{(2)} = R_{ij}^{(1)} + R_{i2}^{(1)} (R_{22}^{(1)})^* R_{2j}^{(1)}$$

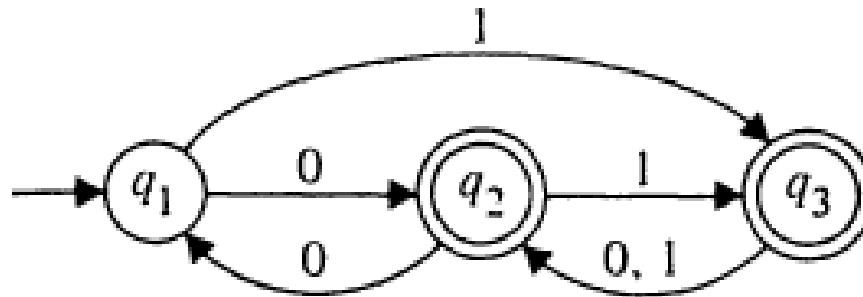
| $R_{11}^{(2)}$ | $1^* + 1^*0(\epsilon + 0 + 1)^*\emptyset$ |
|----------------|---|
| $R_{12}^{(2)}$ | $1^*0 + 1^*0(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$ |
| $R_{21}^{(2)}$ | $\emptyset + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*\emptyset$ |
| $R_{22}^{(2)}$ | $\epsilon + 0 + 1 + (\epsilon + 0 + 1)(\epsilon + 0 + 1)^*(\epsilon + 0 + 1)$ |

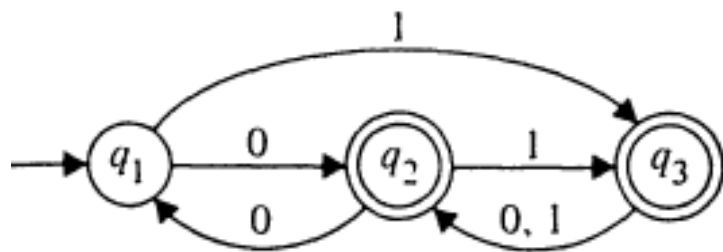
| $R_{11}^{(2)}$ | 1^* |
|----------------|-----------------|
| $R_{12}^{(2)}$ | $1^*0(0 + 1)^*$ |
| $R_{21}^{(2)}$ | \emptyset |
| $R_{22}^{(2)}$ | $(0 + 1)^*$ |

$$R_{12}^{(2)} = 1^*0(0 + 1)^*$$

Ejercicio

- Obtener la exp reg del siguiente autómata





| | $k = 0$ | $k = 1$ | $k = 2$ |
|------------|---------------------------|----------------------------|-----------------------------------|
| r_{11}^k | ϵ | ϵ | $(00)^*$ |
| r_{12}^k | 0 | 0 | $0(00)^*$ |
| r_{13}^k | 1 | 1 | 0^*1 |
| r_{21}^k | 0 | 0 | $0(00)^*$ |
| r_{22}^k | ϵ | $\epsilon + 00$ | $(00)^*$ |
| r_{23}^k | 1 | $1 + 01$ | 0^*1 |
| r_{31}^k | \emptyset | \emptyset | $(0 + 1)(00)^*0$ |
| r_{32}^k | $0 + 1$ | $0 + 1$ | $(0 + 1)(00)^*$ |
| r_{33}^k | ϵ | ϵ | $\epsilon + (0 + 1)0^*1$ |

$$r_{12}^3 = r_{13}^2(r_{33}^2)^*r_{32}^2 + r_{12}^2$$

$$= \mathbf{0^*1}(\epsilon + (\mathbf{0 + 1})\mathbf{0^*1})^*(\mathbf{0 + 1})(00)^* + \mathbf{0(00)^*}$$

$$= \mathbf{0^*1((0 + 1)0^*1)^*(0 + 1)(00)^* + 0(00)^*}$$

$$r_{13}^3 = r_{13}^2(r_{33}^2)^*r_{33}^2 + r_{13}^2$$

$$= \mathbf{0^*1}(\epsilon + (\mathbf{0 + 1})\mathbf{0^*1})^*(\epsilon + (\mathbf{0 + 1})\mathbf{0^*1}) + \mathbf{0^*1}$$

$$= \mathbf{0^*1((0 + 1)0^*1)^*}$$

$$r_{12}^3 + r_{13}^3 = \mathbf{0^*1((0 + 1)0^*1)^*(\epsilon + (0 + 1)(00)^*) + 0(00)^*}.$$

Algebra de expresiones regulares

$$1) r + \emptyset \equiv \emptyset + r \equiv r$$

$$2) r \cdot \varepsilon \equiv \varepsilon \cdot r \equiv r$$

$$3) r \cdot \emptyset \equiv \emptyset \cdot r \equiv \emptyset$$

$$4) r + s \equiv s + r$$

$$5) (r + s) + t \equiv r + (s + t)$$

$$6) (r \cdot s) \cdot t \equiv r \cdot (s \cdot t)$$

$$7) r \cdot (s + t) \equiv r \cdot s + r \cdot t$$

$$8) (s + t) \cdot r \equiv s \cdot r + t \cdot r$$

$$9) r + r \equiv r$$

$$10) \emptyset^* \equiv \varepsilon$$

$$11) r \cdot r^* \equiv r^* \cdot r$$

$$12) r \cdot r^* + \varepsilon \equiv r^*$$

$$13) (r^* \cdot s^*)^* \equiv (r + s)^*$$

$$14) (r^*)^* \equiv r^*$$

$$\bullet (\epsilon + R)^* = R^*$$

$$\bullet R + RS^* = RS^*$$

$$\bullet \emptyset R = R\emptyset = \emptyset$$

$$\bullet \emptyset + R = R + \emptyset = R$$

Ejercicio

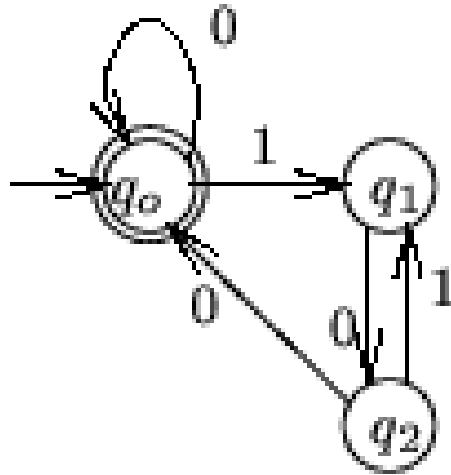
- Simplificar: $(0 + 0^*)1$

Solución

$$\begin{aligned}(0 + 0^*) \cdot 1 &\equiv (0 \cdot \varepsilon + 0 \cdot 0^* + \varepsilon) \cdot 1 \\ &\equiv (0 \cdot (\varepsilon + 0^*) + \varepsilon) \cdot 1 \\ &\equiv (0 \cdot (\varepsilon + 0 \cdot 0^* + \varepsilon) + \varepsilon) \cdot 1 \\ &\equiv (0 \cdot (0 \cdot 0^* + \varepsilon) + \varepsilon) \cdot 1 \\ &\equiv (0 \cdot 0^* + \varepsilon) \cdot 1 \\ &\equiv 0^* \cdot 1\end{aligned}$$

Ejercicio

- Verifique: $\alpha_{q_0, q_0}^X = 0^* + 0^*1(01 + 000^*1)^*000^*$



Gramáticas formales

- La representación de los lenguajes regulares se fundamenta en la noción de gramática formal.
- Intuitivamente, una gramática es un conjunto de reglas para formar correctamente las frases de un lenguaje; así tenemos la gramática del español, del francés, etc.
- La formalización se debe a N. Chomsky y se basa en reglas gramaticales.

- Una regla es una expresión de la forma $\alpha \rightarrow \beta$, en donde tanto α como β son cadenas de símbolos en donde pueden aparecer tanto elementos del alfabeto Σ como unos nuevos símbolos, llamados variables.
- Los símbolos que no son variables son constantes.
- Por ejemplo,
 - una posible regla gramatical es $X \rightarrow aX$.
- La aplicación de una regla $\alpha \rightarrow \beta$ a una palabra $u\alpha v$ produce la palabra $u\beta v$.
- En consecuencia, las reglas de una gramática pueden ser vistas como reglas de reemplazo.
- Por ejemplo,
 - si tenemos una cadena de símbolos $bbXa$, le podemos aplicar la regla $X \rightarrow aX$, dando como resultado la nueva cadena $bbaXa$.

Gramáticas regulares

- Son gramáticas cuyas reglas son de la forma $A \rightarrow aB$ o bien $A \rightarrow a$, donde A y B son variables (no terminales), y a es un símbolo terminal.
- Ejemplo:
 1. $S \rightarrow aA$
 2. $S \rightarrow bA$
 3. $A \rightarrow aB$
 4. $A \rightarrow bB$
 5. $A \rightarrow a$
 6. $B \rightarrow aA$
 7. $B \rightarrow bA$

Definición gramática regular

- Definición.- Una gramática regular es un cuádruplo (V, Σ, R, S) en donde:
 - V es un alfabeto de variables,
 - Σ es un alfabeto de constantes,
 - R , el conjunto de reglas, es un subconjunto finito de $V \times (\Sigma V \cup \Sigma)$.
 - S , el símbolo inicial, es un elemento de V .
- Por ejemplo, la gramática anterior se representaría formalmente como:
- $(\{S, A, B\}, \{a, b\}, \{(S, aA), (S, bA), (A, aB), (A, bB), (A, a), (B, aA), (B, bA)\}, S)$,
- Normalmente las reglas no se escriben como par, sino como $S \rightarrow aA$

Derivación

- Una cadena uXv deriva en un paso una cadena $u\alpha v$, escrito como $uXv \Rightarrow u\alpha v$, si hay una regla $X \rightarrow \alpha \in R$ en la gramática.
- Una cadena $w \in \Sigma^*$ (esto es, formada exclusivamente por constantes) es derivable a partir de una gramática G si existe una secuencia de pasos de derivación $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow w$.
- A una secuencia de pasos de derivación le llamamos simplemente derivación.
- Dicho de otra manera, una palabra $w \in \Sigma^*$ es derivable a partir de G si y sólo si $S \xRightarrow{*} w$.

Lenguaje

- Definición.- El lenguaje generado por una gramática G , $L(G)$, es igual al conjunto de las palabras derivables a partir de su símbolo inicial.

Esto es, $L(G) = \{w \in \Sigma^* \mid S \xRightarrow{*} w\}$.

Ejercicio

- Determine el lenguaje generado por la gramática:

1. $S \rightarrow aA$

2. $S \rightarrow bA$

3. $A \rightarrow aB$

4. $A \rightarrow bB$

5. $A \rightarrow a$

6. $B \rightarrow aA$

7. $B \rightarrow bA$

- El lenguaje generado es el de las palabras en $\{a,b\}$ de longitud par terminadas en a .

Ejercicio

- Proponer una gramática que genere el lenguaje de las palabras en $\{a,b\}$ que contienen la subcadena bb , como abb , $ababba$, etc.
- Propuesta: podemos usar variables para “recordar” situaciones.
 - A, que recuerda que aún no se produce ninguna b .
 - B, que recuerda que se produjo una b .
 - C, que recuerda que ya se produjeron las dos b 's.

Propuesta de solución

1. $A \rightarrow aA$

2. $A \rightarrow bB$

3. $B \rightarrow aA$

4. $B \rightarrow bC$

5. $B \rightarrow b$

6. $C \rightarrow aC$

7. $C \rightarrow bC$

8. $C \rightarrow a$

9. $C \rightarrow b$

Ejercicio

- Proponer una gramática que genere el lenguaje $L = \{a^{2^n} \mid n \geq 0\}$

Solución

- Proponer una gramática que genere el lenguaje $L = \{a^{2^n} \mid n \geq 0\}$
 - $S \rightarrow aA$
 - $S \rightarrow \varepsilon$
 - $A \rightarrow aB$
 - $A \rightarrow a$
 - $B \rightarrow aA$

Gramática regular a AFD

- Teorema. La clase de los lenguajes generados por alguna gramática regular es exactamente la de los lenguajes regulares.
- La prueba consiste en proponer un procedimiento para, a partir de una gramática dada, construir un autómata finito, y viceversa.
- El procedimiento es simple, y consiste en asociar a los símbolos no terminales de la gramática (las variables) los estados de un autómata. Así, para cada regla $A \rightarrow bC$ en la gramática tenemos una transición $\delta(A, b) = C$ en el autómata.
- Sin embargo, queda pendiente el caso de las reglas $A \rightarrow b$. Para estos casos, se tienen transiciones $\delta(A, b) = Z$, donde Z es un nuevo estado para el que no hay un no terminal asociado; Z es un estado final del autómata.

Ejemplo

- Obtener un autómata finito para la gramática regular G siguiente

1. $S \rightarrow aA$

2. $S \rightarrow bA$

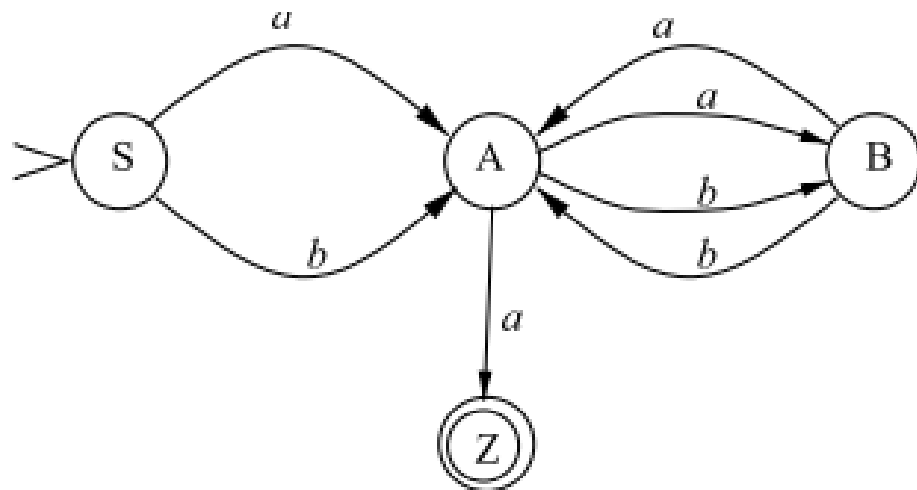
3. $A \rightarrow aB$

4. $A \rightarrow bB$

5. $A \rightarrow a$

6. $B \rightarrow aA$

7. $B \rightarrow bA$



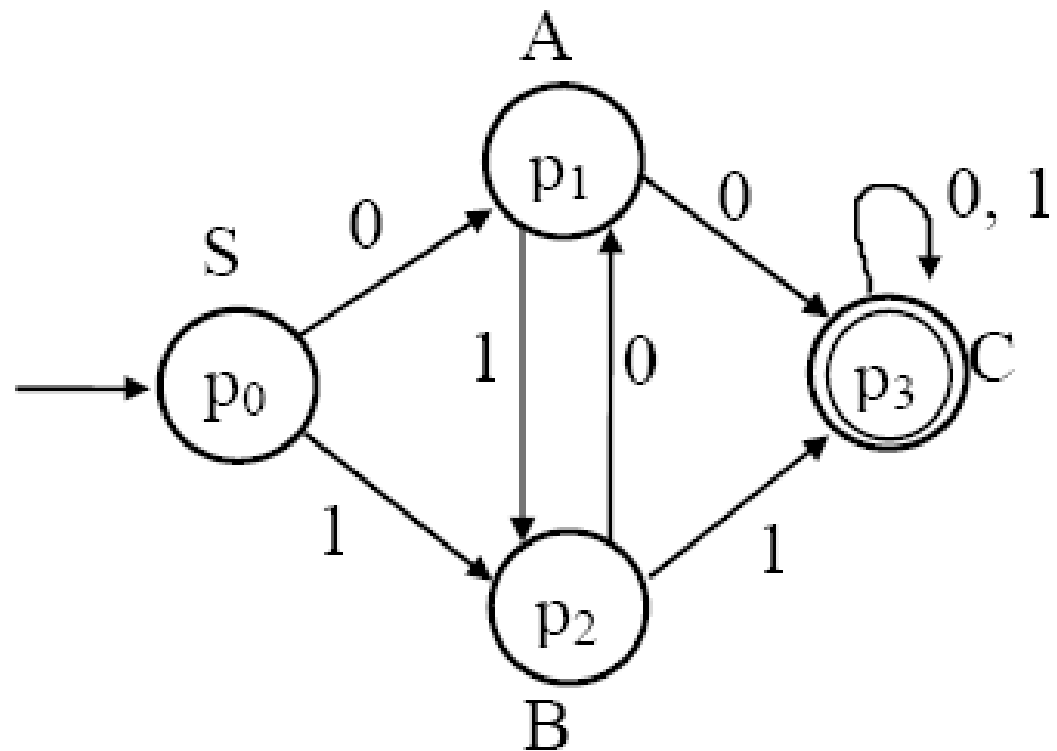
Algoritmo para obtener la gramática regular desde el autómata finito

1. Asociar al estado inicial el símbolo S .
2. Asociar a cada estado del autómata (menos el estado inicial) un símbolo no terminal. Si al estado inicial llega algún arco asociar también un símbolo no terminal (además del símbolo S). No asociar símbolo no terminal a aquellos estados finales de los que no salen arcos.
3. Para cada transición definida $\delta(e_i, a) = e_j$, agregar al conjunto de producciones, la producción $A \rightarrow aB$, siendo A y B los símbolos no terminales asociados a e_i y e_j respectivamente. Si e_j es un estado final, agregar también la producción $A \rightarrow a$. Si e_j es el estado inicial (tiene dos símbolos asociados, el S y un no terminal), utilizar el símbolo no terminal (de esta manera se evita que el símbolo S aparezca a la derecha de una producción).
4. Si el estado inicial es también final agregar la producción $S \rightarrow \varepsilon$.

Ejemplo

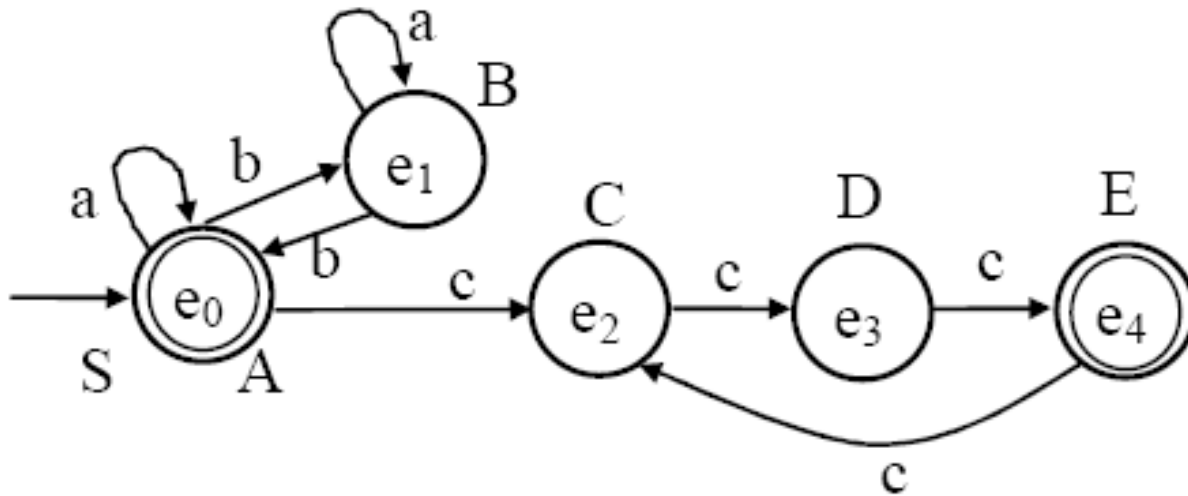
- $L4 = \{ x \mid x \in \{0, 1\}^* \text{ y } x \text{ contiene la subcadena } 00 \text{ ó } x \text{ contiene la subcadena } 11 \}$

- ❑ $S \rightarrow 0A$
- ❑ $S \rightarrow 1B$
- ❑ $A \rightarrow 0C$
- ❑ $A \rightarrow 0$
- ❑ $A \rightarrow 1B$
- ❑ $B \rightarrow 0A$
- ❑ $B \rightarrow 1C$
- ❑ $B \rightarrow 1$
- ❑ $C \rightarrow 0C$
- ❑ $C \rightarrow 0$
- ❑ $C \rightarrow 0$
- ❑ $C \rightarrow 1C$
- ❑ $C \rightarrow 1$

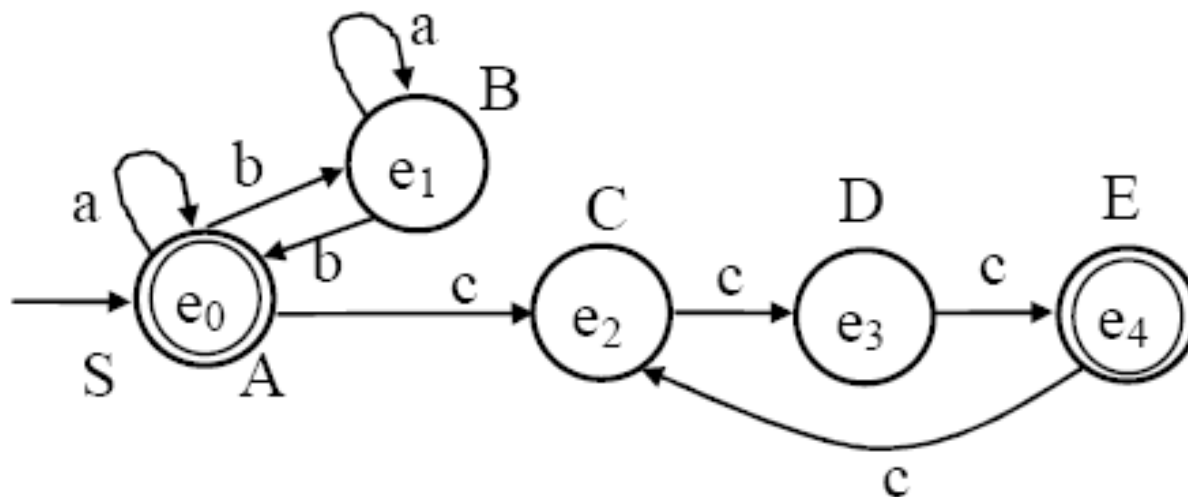


Ejercicio

- $L3 = \{xc^{3m} \mid x \in \{a, b\}^* \text{ y la cantidad de } b\text{'s es par y } m \geq 0\}$



Solución



$S \rightarrow \varepsilon$ (el estado inicial es también final)

$S \rightarrow aA$

$S \rightarrow a$

$S \rightarrow bB$

$S \rightarrow cC$

$A \rightarrow aA$

$A \rightarrow a$

$A \rightarrow cC$

$B \rightarrow aB$

$B \rightarrow bA$ (se usa el símbolo no terminal asociado al estado inicial)

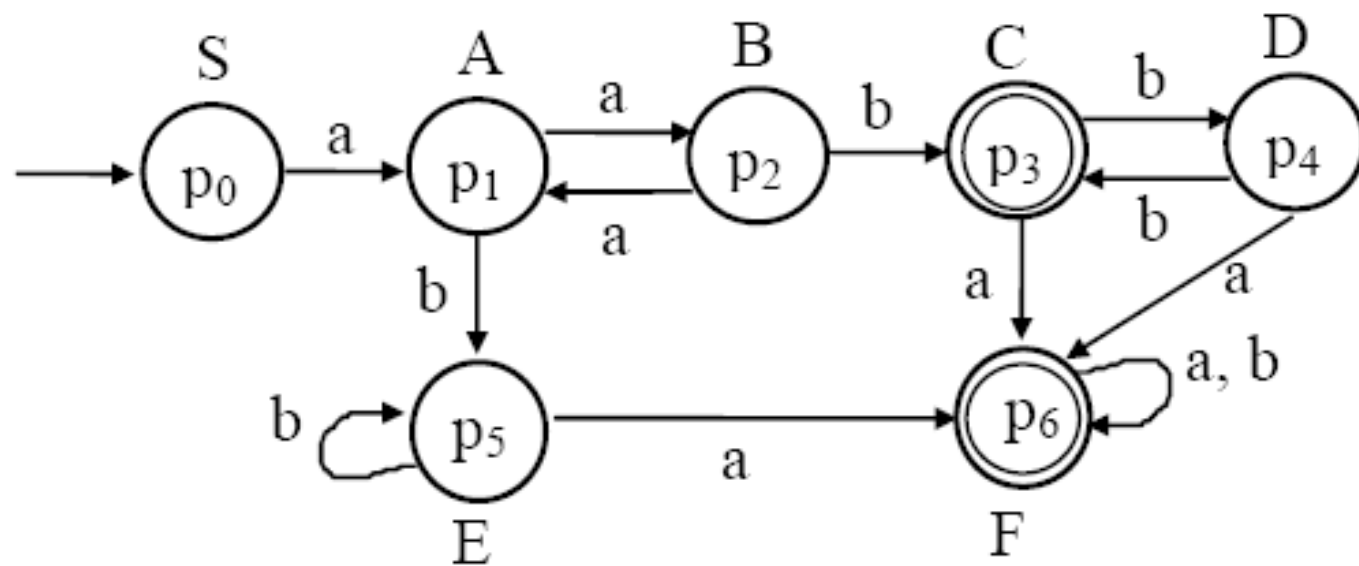
$B \rightarrow b$

$C \rightarrow cD$

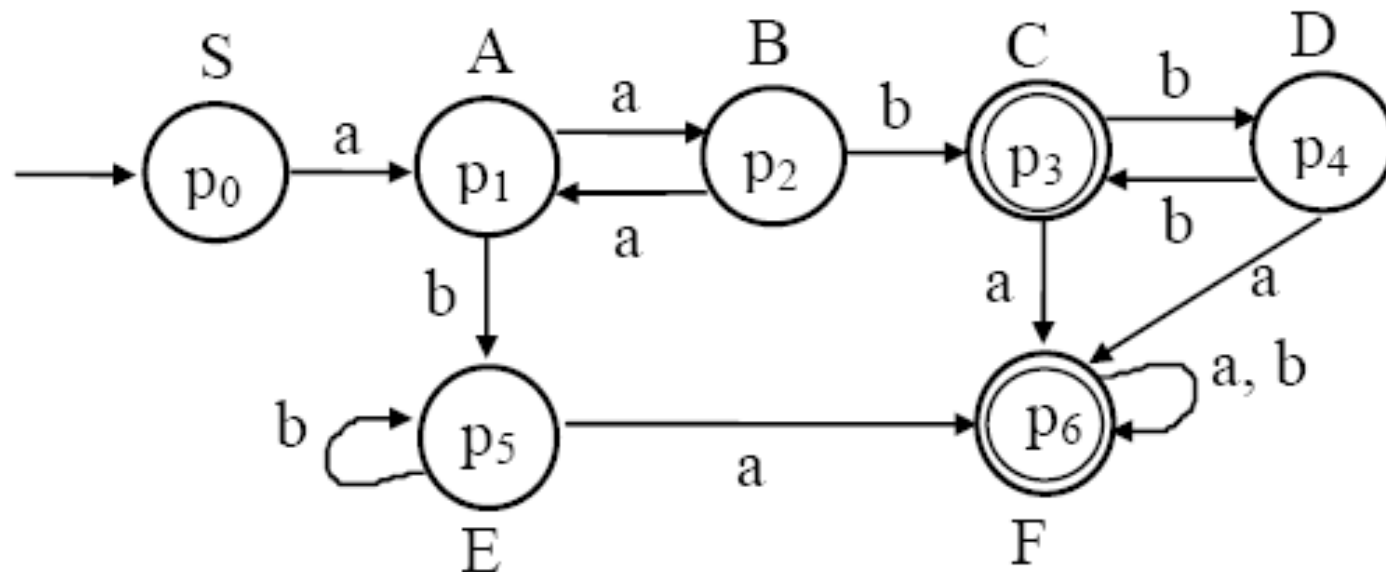
$D \rightarrow cE$

$D \rightarrow c$

Ejercicio



Solución



$S \rightarrow aA$

$A \rightarrow aB$

$A \rightarrow bE$

$B \rightarrow aA$

$B \rightarrow bC$

$B \rightarrow b$

$C \rightarrow bD$

$C \rightarrow aF$

$C \rightarrow a$

$D \rightarrow bC$

$D \rightarrow b$

$D \rightarrow aF$

$D \rightarrow a$

$E \rightarrow bE$

$E \rightarrow aF$

$E \rightarrow a$

$F \rightarrow aF$

$F \rightarrow a$

$F \rightarrow bF$

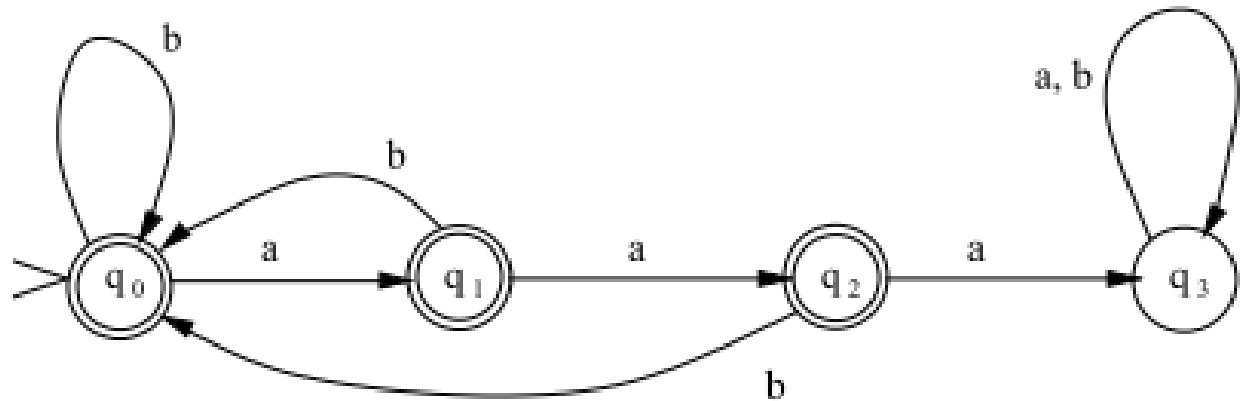
$F \rightarrow b$

Ejercicio

- Construye el AFD que acepta palabras que no contienen 3 a's seguidas.
 - Proponer una gramática regular.
-

Solución

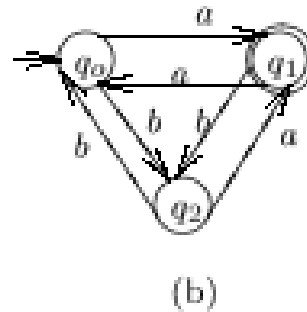
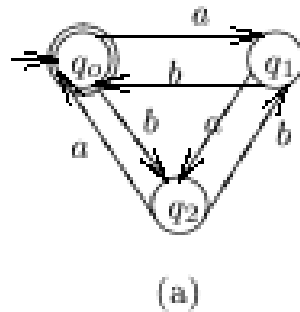
- AFD que acepta palabras que no contienen 3 a's seguidas.



- | | |
|----------------------------|------------------------------------|
| 1.- $Q_0 \rightarrow aQ_1$ | 8.- $Q_3 \rightarrow bQ_3$ |
| 2.- $Q_0 \rightarrow bQ_0$ | 9.- $Q_0 \rightarrow a$ |
| 3.- $Q_1 \rightarrow aQ_2$ | 10.- $Q_0 \rightarrow b$ |
| 4.- $Q_1 \rightarrow bQ_0$ | 11.- $Q_1 \rightarrow a$ |
| 5.- $Q_2 \rightarrow aQ_3$ | 12.- $Q_1 \rightarrow b$ |
| 6.- $Q_2 \rightarrow bQ_0$ | 13.- $Q_2 \rightarrow b$ |
| 7.- $Q_3 \rightarrow aQ_3$ | 14.- $Q_0 \rightarrow \varepsilon$ |

Ejercicios

- Describa cuáles son los conjuntos de cadenas denotadas por las siguientes expresiones regulares:
 - $(11+0)^*(00+1)^*$
 - $(1+01+001)^*(\epsilon+0+00)^*$
 - $(00+11+(01+10)(00+11)^*)^*$
- Construya los correspondientes autómatas que aceptan los mismos lenguajes que las siguientes expresiones regulares:
 - $10+(0+11)0^*1$
 - $01(((10)^*+111)^*+0)^*1$
- Encontrar la ER asociada a cada uno de los AFD que se muestran en la figura:



Limitaciones de los AF

- Los AF están limitados a los estados de que disponen como único medio para “recordar” la serie de símbolos recibidos hasta un momento dado.
- Puesto de otra manera, debemos considerar que, en un AF, la única traza de los símbolos recibidos es el estado en que se encuentra.
- Por lo mismo, varias secuencias distintas de caracteres que llevan a un mismo estado son consideradas como indistinguibles.

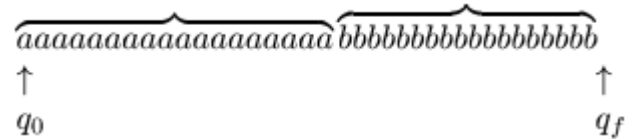
Ejemplo

- Para el lenguaje $\{a^n b^n\}$ no es posible construir un autómata finito que lo acepte, ni representarlo por una expresión regular o gramática regular.
- En efecto, supongamos que un AFD está recorriendo una palabra $a^n b^n$, entonces al terminar el grupo de a 's el autómata debe recordar cuántas encontró, para poder comparar con el número de b 's.
- Ahora bien, como la cantidad de a 's que puede haber en la primera mitad de la palabra es arbitraria, dicha cantidad no puede recordarse con una cantidad de memoria fija, como es la de los autómatas finitos.

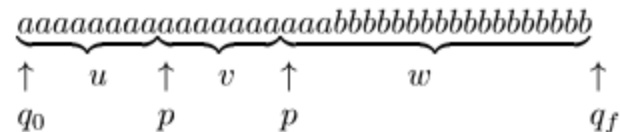
Ejemplo de un conjunto no regular

- $B = \{a^n b^n \mid n \geq 0\} = \{\epsilon, ab, aabb, aaabbb, \dots\}$
 - El conjunto de todas las cadenas de la forma a^*b^* con exactamente igual número de a's que de b's.
- Demostrar que no es regular. Dem por contradicción.
 - Asumimos que B es regular, significa que existe un AF $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ tal que $L(M) = B$. Sea $k = |Q|$.

- Considérese la acción de M ante la entrada $a^n b^n$, para $n \gg k$ (n mucho mayor que k , $n = k + 1$). M inicia en el estado q_0 , dado que la cadena $a^n b^n$ está en B , M debe aceptarla; así que M debe alcanzar algún estado final q_f después de examinar $a^n b^n$:



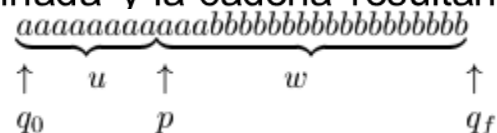
- Dado que $n \gg k$, debe existir algún estado p por el cual M necesariamente debe pasar más de una vez mientras está examinando la secuencia inicial de a 's. Esto significa que podemos dividir la cadena $a^n b^n$ en tres subcadenas u , v y w , en donde v es la cadena de a 's examinadas entre dos ocurrencias del estado p :



- Sea $j = |v| > 0$. para éste ejemplo $j = 7$. Podemos apreciar que

$$\begin{aligned} \hat{\delta}(q_0, u) &= p \\ \hat{\delta}(p, v) &= p \\ \hat{\delta}(p, w) &= q_f \end{aligned}$$

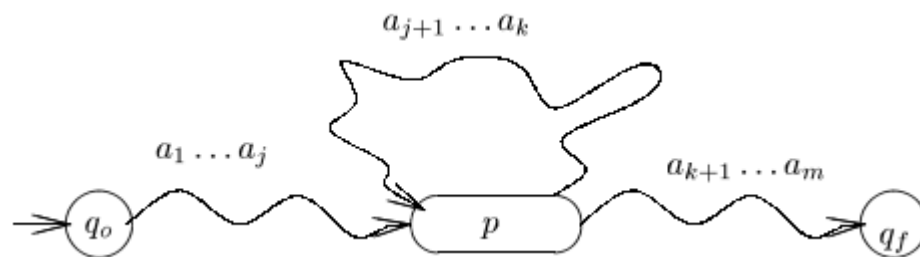
- La cadena v puede ser eliminada y la cadena resultante sería aceptada (error)



- Claramente, tal cadena tiene un número estrictamente menor de a 's que de b 's: $uw = a^{n-j} b^n \in L(M)$ pero $uw \notin B$ lo cual es una contradicción de la premisa $L(M) = B$.

- Además, podemos insertar cualquier número de copias que queramos de la subcadena v y la cadena que resulte también sería aceptada por M : $uv^3w = a^{n+2j}b^n \in L(M)$:

$$\begin{aligned}
 \hat{\delta}(q_0, uvvvw) &= \hat{\delta}(\hat{\delta}(\hat{\delta}(\hat{\delta}(\hat{\delta}(q_0, u), v), v), v), w) \\
 &= \hat{\delta}(\hat{\delta}(\hat{\delta}(\hat{\delta}(p, v), v), v), w) \\
 &= \hat{\delta}(\hat{\delta}(\hat{\delta}(p, v), v), w) \\
 &= \hat{\delta}(\hat{\delta}(p, v), w) \\
 &= \hat{\delta}(p, w) = q_f
 \end{aligned}$$



Lema de Sondeo (Pumping)

- Útil para demostrar que un conjunto no es regular. La idea es que siempre que un AF examine una cadena larga (mayor que el número de estados) y la acepte entonces significa que debe haber un estado repetido (pasa por el más de una vez) y se pueden insertar copias de la subcadena entre las dos ocurrencias de ése estado repetido de tal forma que la cadena resultante también es aceptada.

Teorema Lema del Sondeo

- (P) Sea A un conjunto regular. A cumple con la siguiente propiedad:
 - Existe un $k \geq 0$ tal que para cualquiera cadenas x, y, z con $xyz \in A$ y $|y| \geq k$, existen cadenas u, v, w tales que $y = uvw$, $v \neq \epsilon$, y para toda $i \geq 0$ la cadena $xuv^i w z \in A$

- (Forma Contrapositiva). Sea A un conjunto regular. Supóngase que
 - Para toda $k \geq 0$ existen cadenas x, y, z tales que $xyz \in A$ y $|y| \geq k$; y para toda cadena u, v, w con $y = uvw$, $v \neq \epsilon$, y existe $i \geq 0$ tal que la cadena $xuv^i wz \notin A$
 - *Por lo tanto A no es regular*

Ejemplo

- Aplicaremos el lema del sondeo para demostrar que el conjunto $A = \{a^n b^m \mid n \geq m\}$ no es regular.
- Suponemos que es regular, elegimos algún núm k .
- Además, $x=a^k$, $y=b^k$ y $z= \varepsilon$. Entonces $xyz = a^k b^k \in A$ con $|y|=k$.
- Sea u, v, w donde $|u|=j$, $|v|=m > 0$ y $|w|=n$ ($k=j+m+n$) tales que $y=uvw$ y $v \neq \varepsilon$. Si $i=2$,

$$\begin{aligned}xuv^2wz &= a^k b^j b^m b^m b^n \\&= a^k b^{j+2m+n} \\&= a^k b^{k+m} \notin A\end{aligned}$$

- El número de b's es estrictamente mayor al número de

Ejercicio

- ¿Cuál de los siguientes lenguajes es un conjunto regular? Justifique su respuesta.

1. $\{0^{2n} \mid n \geq 1\}$.

2. $\{0^m 1^n 0^{m+n} \mid n \geq 1, m \geq 1\}$.

3. $\{0^n \mid n \text{ es primo}\}$.