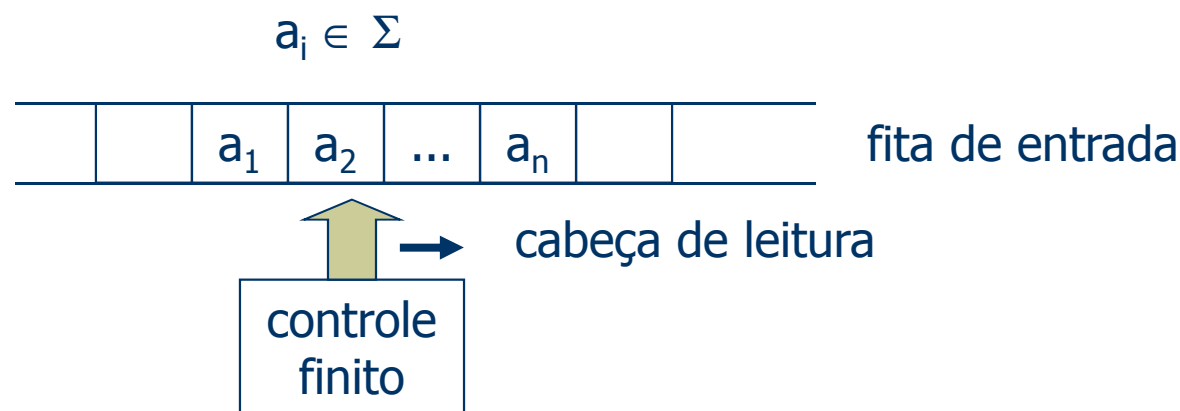


1.2 - Autômatos Finitos e Linguagens Regulares

- ♦ Definição: Um autômato de estados finitos é uma quintupla $A = (Q, \Sigma, \delta, q_0, F)$ onde:
 - Q - conjunto finito não vazio (estados)
 - Σ - alfabeto (de entrada); $\Sigma \cap Q = \emptyset$
 - $q_0 \in Q$ (estado inicial)
 - $F \subseteq Q$ (conjunto de estados finais)
 - $\delta : Q \times \Sigma \rightarrow Q$ (função de transição de estados)

Simbolicamente:



$$\delta(q, a) = q'$$

autômato estando no estado q e lendo o símbolo a na fita de entrada, move a cabeça leitora uma posição para a direita e vai para o estado q' .

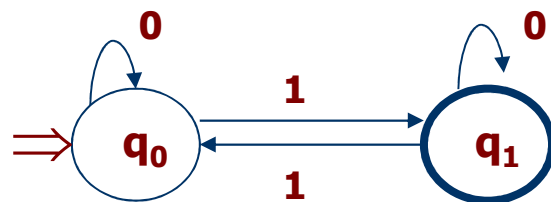
Autômatos Finitos e Linguagens Regulares

- ♦ Função δ estendida: $\delta : Q \times \Sigma^* \rightarrow Q$
 $\delta(q, \Lambda) = q$
 $\delta(q, xa) = \delta(\delta(q, x), a) \quad x \in \Sigma^*, a \in \Sigma$

$\delta(q, x) = q'$ autômato estando no estado q e lendo o símbolo mais à esquerda de x , vai estar no estado q' após todos os símbolos de x terem sido lidos.

- ♦ Definição: Sejam A um autômato finito e $x \in \Sigma^*$. A cadeia x é aceita por A se $\delta(q_0, x) \in F$.
- ♦ Definição: Seja A um autômato finito. A linguagem reconhecida por A é:
$$L(A) = \{x \in \Sigma^* \mid \delta(q_0, x) \in F\}$$

Exemplo:



(diagrama de estados)

$$A = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$$

$$L(A) = \{x \in \Sigma^* \mid x \text{ contém número ímpar de } 1\text{'s}\}$$

Relações de Equivalência e Autômatos Finitos

- ♦ Definição: Uma relação binária R sobre um conjunto S é um conjunto de pares de elementos de S .
- ♦ Notação: se $(a,b) \in R$ então $a R b$
- ♦ Definição: Uma relação binária R sobre S é uma relação de equivalência se:
 - a) R é reflexiva ($s R s, \forall s \in S$)
 - b) R é simétrica ($s, t \in S, s R t \Rightarrow t R s$)
 - c) R é transitiva ($s, t, u \in S, s R t, t R u \Rightarrow s R u$)
- ♦ Teorema: Se R é uma relação de equivalência sobre S então é possível dividir S em k subconjuntos distintos ($k \geq 1$) chamados classes de equivalência tais que $a R b \Leftrightarrow a$ e b pertencem ao mesmo subconjunto (**em outras palavras: uma relação de equivalência R sobre S induz uma partição de S**).
- ♦ Prova: Seja $[x] = \{ y \mid x R y \}$. O teorema garante que, para $\forall a, b \in S$:
 - a) $[a] = [b]$, ou
 - b) $[a] \cap [b] = \emptyset$

Vamos admitir, por absurdo, que isto não vale. Temos então:

$$c \in [a]$$

$$\Rightarrow a R c$$

$$c \in [b]$$

$$\Rightarrow b R c \Rightarrow (\text{simetria}) c R b \Rightarrow (\text{transitividade}) a R b$$

$$d \in [b]$$

$$\Rightarrow b R d \Rightarrow (\text{transitividade}) a R d \Rightarrow d \in [a]. \text{ CONTRADIÇÃO!}$$

$$d \notin [a]$$

Relações de Equivalência e Autômatos Finitos

- ♦ Exemplo: R sobre $N = \{1,2,3,\dots\}$ tal que $i R j \Leftrightarrow |i-j|$ é divisível por 3.
Classes de equivalência:
 $c_1 = \{ 1, 4, 7, \dots \}$
 $c_2 = \{ 2, 5, 8, \dots \}$
 $c_3 = \{ 3, 6, 9, \dots \}$
- ♦ Definição: O índice de uma relação de equivalência R , representado por $i(R)$, é o número de classes de equivalência de R .
- ♦ Definição: Sejam R_1 e R_2 , relações de equivalência. R_1 refina R_2 se $R_1 \subseteq R_2$ (isto é, $x R_1 y \Rightarrow x R_2 y$).
- ♦ Exemplo: R_1 e R_2 sobre $N = \{1,2,3,\dots\}$ tais que:
 $i R_1 j \Leftrightarrow |i-j|$ é divisível por 6
 $i R_2 j \Leftrightarrow |i-j|$ é divisível por 3
Classes de equivalência de R_1 :
 $c_1 = \{ 1, 7, 13, \dots \}$
 $c_2 = \{ 2, 8, 14, \dots \}$
 $c_3 = \{ 3, 9, 15, \dots \}$
 $c_4 = \{ 4, 10, 16, \dots \}$
 $c_5 = \{ 5, 11, 17, \dots \}$
 $c_6 = \{ 6, 12, 18, \dots \}$
 $x R_1 y \Rightarrow x R_2 y$
 R_1 refina R_2

Exercício: Mostrar que se R_1 refina R_2 então $i(R_1) \geq i(R_2)$.

Relações de Equivalência e Autômatos Finitos

- Definição: Seja R uma relação de equivalência sobre Σ^* . R é uma relação de congruência à direita se $x R y \Rightarrow (\forall z \in \Sigma^*) xz R yz$.

- Exemplo: Seja R sobre Σ^* definida por: $x R y \Leftrightarrow \delta(q_0, x) = \delta(q_0, y)$

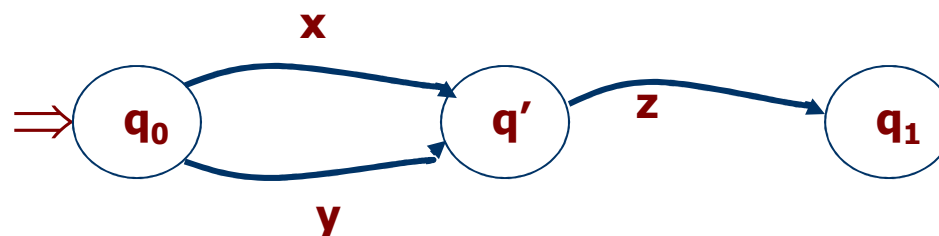
Exercício: Mostrar que R é uma relação de equivalência.

Vamos mostrar que R é uma relação de congruência à direita.

$$x R y \Rightarrow \delta(q_0, x) = \delta(q_0, y) = q'$$

$$\text{Mas, para } \forall z \in \Sigma^*: \delta(q_0, xz) = \delta(\delta(q_0, x), z) = \delta(q', z) = \delta(\delta(q_0, y), z) = \delta(q_0, yz).$$

(intuitivamente: a relação R é a resposta do autômato à cadeia).



- Definição: Sejam R , relação de equivalência sobre Σ^* , e $L \subseteq \Sigma^*$. R refina L se: $x R y \Rightarrow (x \in L \Leftrightarrow y \in L)$. Em outras palavras: se R refina L , L é a união de algumas (ou todas) as classes de equivalência de R .
- Definição: Seja $L \subseteq \Sigma^*$. A relação R_L é definida por:
$$x R_L y \Leftrightarrow (\forall z \in \Sigma^*) (xz \in L \Leftrightarrow yz \in L).$$

Relações de Equivalência e Autômatos Finitos

- ♦ Teorema:

- a) R_L é uma relação de congruência à direita
- b) R_L refina L
- c) se R é uma relação de congruência à direita que refina L então $R \subseteq R_L$.

Em outras palavras, R_L é a menor (menos classes de equivalência) relação de congruência à direita que refina L .

- ♦ Prova: Exercício!

- ♦ Teorema (de Nerode): Seja $L \subseteq \Sigma^*$. São equivalentes:

- a) L é regular.
- b) Existe uma relação de congruência à direita R sobre Σ^* que refina L e de índice finito.
- c) $i(R_L)$ é finito.

- ♦ Prova:

(a) \Rightarrow (b). L é regular \Rightarrow Existe autômato finito $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L = L(A)$ (isso será mostrado num teorema mais à frente). Seja R uma relação definida por: $x R y \Leftrightarrow \delta(q_0, x) = \delta(q_0, y)$. Como vimos, R é relação de congruência à direita e (obviamente!) refina L . Se x e y estão na mesma classe de equivalência ($x R y$) então $\delta(q_0, x) = \delta(q_0, y)$. Como o autômato tem um número finito de estados, $i(R) \leq \#Q$. Logo $i(R)$ é finito.

Relações de Equivalência e Autômatos Finitos

(b) \Rightarrow (c). Pelo teorema anterior, $R \subseteq R_L$. Logo: $i(R) \geq i(R_L)$. Como $i(R)$ é finito, $i(R_L)$ é finito.

(c) \Rightarrow (a). Devemos construir autômato finito $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L = L(A)$. Seja $[x]$ a classe de equivalência de R_L que contém $x \in \Sigma^*$. Então:

$$Q = \{ [x] \mid x \in \Sigma^* \}$$

$$q_0 = [\Lambda]$$

$$F = \{ [x] \mid x \in L \}$$

$\delta([x], a) = [xa]$; $x \in \Sigma^*$, $a \in \Sigma$ (notar que, como R_L é relação de congruência à direita, só existe uma classe que contém xa , $x \in \Sigma^*$, $a \in \Sigma$, ou seja, a definição de δ é consistente).

Temos então: $\delta(q_0, x) = \delta([\Lambda], x) = [\Lambda x] = [x]$.

Portanto: $x \in L(A) \Leftrightarrow [x] \in F \Leftrightarrow x \in L$. Logo: $L = L(A)$.

- ♦ O teorema de Nerode é muito útil para provar que certas classes de linguagens não são regulares.
- ♦ Exemplo: $L = \{ a^i b^j \mid j \geq i \geq 0 \}$. Seja R_L a relação de equivalência de Nerode, ou seja: $x R_L y \Leftrightarrow (\forall z \in \Sigma^*) (xz \in L \Leftrightarrow yz \in L)$. Para mostrar que L não é regular vamos mostrar que $i(R_L)$ não é finito.

Relações de Equivalência e Autômatos Finitos

Seja $P^n = \{ a^{i+n}b^i \mid i \geq 0 \}$ ($n \geq 1$). P^n é o conjunto das cadeias que são prefixo de alguma cadeia de L e que tem n a's a mais do que b's. Seja $P = \cup P^i$ ($i \geq 1$)

Vamos mostrar que as classes de equivalência de R_L são:

- 1) L
- 2) $\Sigma^* - L - P$
- 3) $P^i, i \geq 1$

ou seja, para $x, y \in \Sigma^*$, se $x R_L y$ então:

- a) $x, y \in L$, ou
- b) $x, y \in \Sigma^* - L - P$, ou
- c) $x, y \in P^i$, para algum $i \geq 1$.

- a) se para $z \in \Sigma^*$, $xz \in L \Leftrightarrow yz \in L$, então, em particular, para $z = \Lambda$, $x \in L \Leftrightarrow y \in L$.
- b) $x \in \Sigma^* - L - P$. Então x não é prefixo de qualquer cadeia de L . Portanto, não existe $z \in \Sigma^*$ tal que $xz \in L$. Como $xz \in L \Leftrightarrow yz \in L$, não existe $z \in \Sigma^*$ tal que $yz \in L$. Portanto, $y \in \Sigma^* - L - P$.
- c) Seja $x \in P^i$, ou seja, $x = a^n b^m$ tal que $n-m = i$. Devemos mostrar que $y \in P^i$. Vamos admitir, por absurdo, que $y \notin P^i$. Então três casos podem ocorrer: $y \in L$, ou $y \in \Sigma^* - L - P$, ou $y \in P^j$ com $j \neq i$. Vamos analisar cada um desses casos:

Relações de Equivalência e Autômatos Finitos

- 1) $y \in L$. Como para $\forall z \in \Sigma^*$, $xz \in L \Leftrightarrow yz \in L$, então, em particular, deve valer para $z = \Lambda$. Mas com $z = \Lambda$, $xz \notin L$ e $yz \in L$. Logo: $y \notin L$.
CONTRADIÇÃO!
 - 2) $y \in \Sigma^* - L - P$. Logo, não existe $z \in \Sigma^*$ tal que $yz \in L$. No entanto, como $x \in P_i$, é possível encontrar $z \in \Sigma^*$ tal que $xz \in L$. Logo: $y \notin \Sigma^* - L - P$.
CONTRADIÇÃO!
 - 3) $y \in P^j$ com $j \neq i$. Seja $j < i$. Seja $z = b^j$. Então: $yz = a^n b^{m+j}$ onde $n-m = j$. Logo: $m+j = n$ e portanto, $yz \in L$. Por outro lado, $xz = a^n b^{m+j}$ onde $n-m = i$, ou seja, $n = m + i$. Como $j < i$, por hipótese, $n > m + j$, ou seja, $xz \notin L$. Logo: $y \notin P^j$ com $j \neq i$. CONTRADIÇÃO!
- Dessa forma, provamos que R_L tem infinitas classes de equivalência. Pelo teorema de Nerode, $L = \{ a^i b^j \mid j \geq i \geq 0 \}$ não pode ser regular.

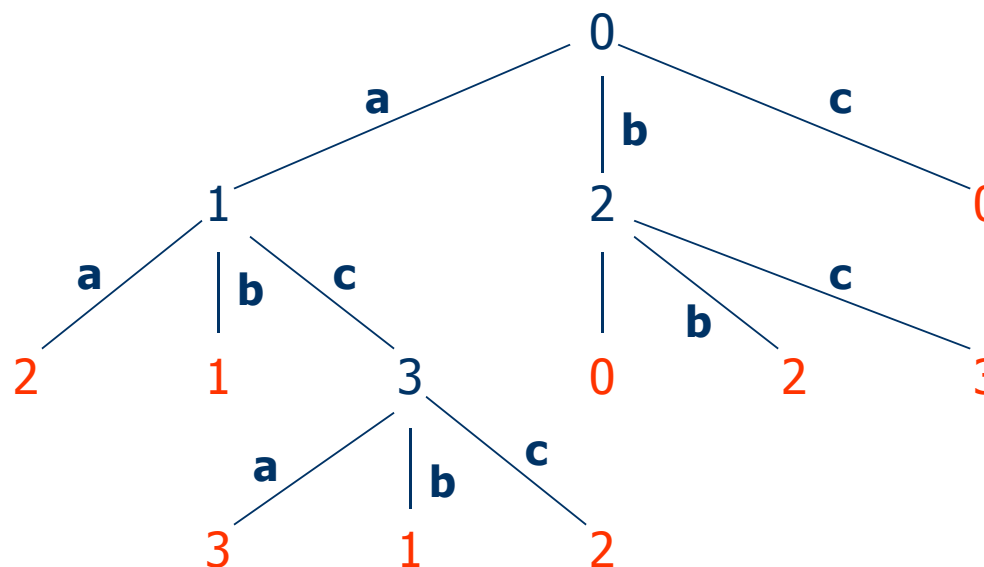
Minimização de Estados de um Autômato Finito

- Definição: Seja $A = (Q, \Sigma, \delta, q_0, F)$. Um estado $q \in Q$ é acessível se $\exists x \in \Sigma^*$ tal que $q = \delta(q_0, x)$.
- Definição: O autômato conexo associado a $A = (Q, \Sigma, \delta, q_0, F)$ é definido por: $A^c = (Q', \Sigma, \delta', q_0, F')$ onde: $Q' = \{ q \in Q \mid q \text{ é acessível} \}$, $F' = F \cap Q'$ e $\delta' = \delta \cap (Q' \times \Sigma \times Q')$.

Minimização de Estados de um Autômato Finito

- ♦ Definição: Dois autômatos A_1 e A_2 são equivalentes se $L(A_1) = L(A_2)$.
- ♦ Teorema: Para todo autômato finito A , A e A^c são equivalentes.
- ♦ Prova: Exercício!
- ♦ A determinação de estados acessíveis pode ser automatizada organizando a função δ em forma de árvore e verificando os estados que ocorrem.
- ♦ **Exemplo**: $A = (Q, \Sigma, \delta, q_0, F)$ com: $\Sigma = \{a, b, c\}$, $Q = \{0,1,2,3,4,5\}$; $q_0 = 0$ e δ dada por:

δ	a	b	c
0	1	2	0
1	2	1	3
2	0	2	3
3	3	1	2
4	5	1	2
5	0	4	5



- ♦ Logo: somente os estados 0,1,2 e 3 são acessíveis.

Minimização de Estados de um Autômato Finito

- ♦ Definição: Dois estados q e q' são equivalentes ($q \equiv q'$) se e somente se:
 $(\forall x \in \Sigma^*) (\delta(q, x) \in F \Leftrightarrow \delta(q', x) \in F)$.
- ♦ **Exercício. Mostrar que \equiv é relação de equivalência.**
- ♦ A partição π (partição de Q induzida pela relação \equiv) permite identificar elementos redundantes em Q (do ponto de vista de reconhecimento da linguagem), pois se $q \equiv q'$ (q e q' na mesma classe de equivalência) não faz diferença se o autômato se encontra no estado q ou no estado q' quando uma cadeia $x \in \Sigma^*$ começar a ser processada. Logo, o autômato pode ser minimizado escolhendo-se apenas um elemento de cada uma das classes de equivalência da relação \equiv .
- ♦ Definição: q é k -equivalente a q' ($q \equiv^k q'$) se e somente se:
 $(\forall x \in \Sigma^*, |x| \leq k \Rightarrow (\delta(q, x) \in F \Leftrightarrow \delta(q', x) \in F))$.
Desta definição segue que se $q \equiv^k q'$ então $q \equiv^m q'$ para todo $m \leq k$, pois:
 $q \equiv^k q' \Rightarrow (\forall x \in \Sigma^*, |x| \leq m \leq k \Rightarrow (\delta(q, x) \in F \Leftrightarrow \delta(q', x) \in F)) \Rightarrow q \equiv^m q'$
Portanto π_k (partição de Q induzida pela relação \equiv^k) refina π_m ($\pi_k \subseteq \pi_m, m \leq k$)
- ♦ Teorema: $q \equiv^{k+1} q' \Leftrightarrow (q \equiv^k q' \text{ e } \delta(q, a) \equiv^k \delta(q', a), \forall a \in \Sigma)$
- ♦ Prova: (\Rightarrow) $q \equiv^{k+1} q'$. Logo: $q \equiv^k q'$ pois $\pi_{k+1} \subseteq \pi_k$. Além disso, para toda cadeia $ax, a \in \Sigma, x \in \Sigma^*$, tal que: $|ax| \leq k+1$, temos: $\delta(q, ax) \in F \Leftrightarrow \delta(q', ax) \in F$ (pois $q \equiv^{k+1} q'$). Logo: $\delta(\delta(q, a), x) \in F \Leftrightarrow \delta(\delta(q', a), x) \in F$. Como $|x| \leq k$, temos que: $\delta(q, a) \equiv^k \delta(q', a)$.
(\Leftarrow) Exercício!

Minimização de Estados de um Autômato Finito

- ♦ O teorema anterior pode ser usado para mostrar (por indução) que se existe k tal que $\pi_k = \pi_{k+1}$ então $\pi_k = \pi_{k+m}$ para todo $m \geq 0$, ou seja, não existem mais refinamentos de π_k . Neste caso, como $\pi \subseteq \pi_k$ temos que $\pi = \pi_k$.

(para uma prova formal ver: BOOTH, T.L. "Sequential machines and automata theory")

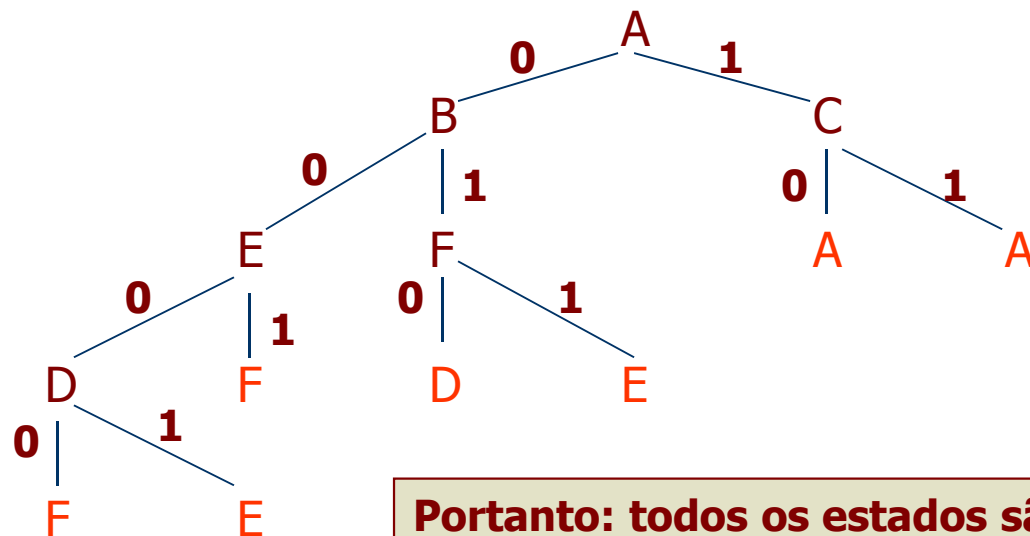
- ♦ Portanto, o teorema anterior garante que podemos encontrar a relação \equiv considerando, sucessivamente, as relações $\equiv^0, \equiv^1, \equiv^2$, e assim por diante até que $\pi_k = \pi_{k+1}$, $k \geq 0$. O problema agora é saber se existe tal k (ou seja, se esse processo pára).
- ♦ Teorema: Seja $A = (Q, \Sigma, \delta, q_0, F)$ com $\#Q = n$ ($n \geq 2$). Então existe k tal que $\pi_k = \pi_{k+1}$ e $k \leq n-2$.
- ♦ Prova: Seja $\#\pi_0 = 1$ (note que $q \equiv^0 q' \Leftrightarrow \delta(q, \Lambda) \in F \Leftrightarrow \delta(q', \Lambda) \in F$, ou seja, que $\#\pi_0 = 1$ quando $\#F = 1$). Neste caso, $\pi_1 = \pi_0$ (**Exercício!**) e $k = 0 \leq n-2$.
Seja, então, $\#\pi_0 \geq 2$. Se $\pi_i \neq \pi_{i+1}$ então $\#\pi_{i+1} > \#\pi_i$. Como cada classe de equivalência de \equiv^i deve conter pelo menos um estado e como $\#Q = n$, então: $2 \leq \#\pi_i < n$. Logo, no pior caso, existe k tal que:
 $2 = \#\pi_0 < \#\pi_1 < \#\pi_2 < \dots < \#\pi_k = \#\pi_{k+1} = n$ e, nesse caso, $k = n-2$.
Portanto, em geral, existe $k \leq n-2$ tal que $\pi_k = \pi_{k+1}$.

Esse teorema, portanto, garante que pode ser construído um algoritmo para determinar a relação \equiv a partir das relações $\equiv^0, \equiv^1, \equiv^2, \dots$

Minimização de Estados de um Autômato Finito

- Exemplo: $A = (\{A,B,C,D,E,F\}, \{0,1\}, \delta, A, \{E,F\})$ com δ dada por:

δ	0	1
A	B	C
B	E	F
C	A	A
D	F	E
E	D	F
F	D	E



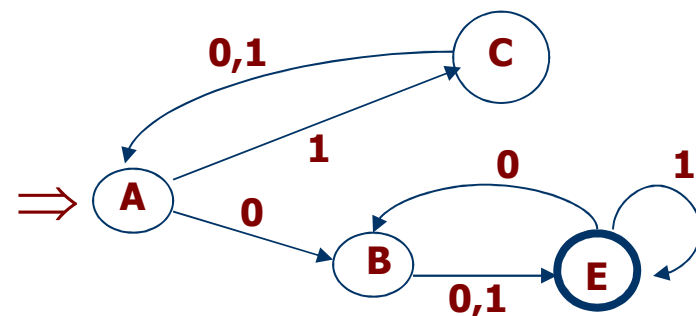
Portanto: todos os estados são acessíveis.

- Construção da relação \equiv

π_0	(A B C D)	(E F)
π_1	(A C)	(B D) (E F)
π_2	(A) (C)	(B D) (E F)
π_3	(A) (C)	(B D) (E F)

Logo: $\pi_3 = \pi$

**autômato mínimo
e equivalente a A**



Autômato Finito Não Determinístico

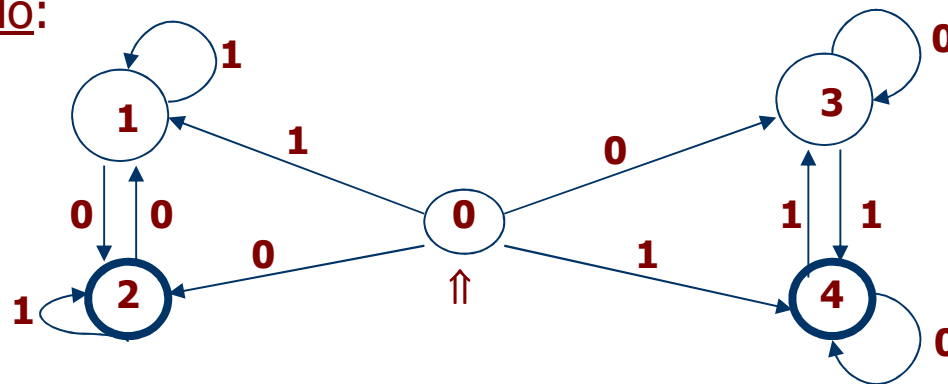
- Definição: Um autômato finito não determinístico (AFND) é uma quintupla $A = (Q, \Sigma, \delta, q_0, F)$ com Q, Σ, q_0 e F definidos como no caso determinístico e δ dada por:

$\delta : Q \times \Sigma \rightarrow \mathbf{P}(Q)$, onde $\mathbf{P}(Q)$ é o conjunto dos subconjuntos de Q (conjunto potência).

$\delta(q,a) = \{q_1, \dots, q_n\} \Leftrightarrow$ o autômato estando no estado q e lendo o símbolo a na fita de entrada, move sua cabeça leitora uma posição para a direita e escolhe qualquer um dos q_i ($i = 1, \dots, n$) como seu próximo estado (conceitualmente, é equivalente a imaginar que o autômato se subdivide em n cópias, cada uma das quais estando em um estado q_i , $i = 1, \dots, n$).

- Definição: Seja $x \in \Sigma^*$. x é aceita por um AFND $A = (Q, \Sigma, \delta, q_0, F)$ se: $\delta(q_0, x) \cap F \neq \emptyset$ (portanto: $L(A) = \{ x \in \Sigma^* \mid \delta(q_0, x) \cap F \neq \emptyset \}$).

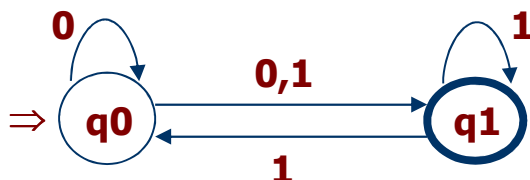
- Exemplo:



Exercício:
Determinar $L(A)$

Autômato Finito Não Determinístico

- ♦ Teorema: Seja L um conjunto aceito por um AFND. Então existe um autômato finito determinístico que aceita L .
- ♦ Prova (Informal):
 $A = (Q, \Sigma, \delta, q_0, F)$ (AFND)
 $A' = (Q', \Sigma, \delta', q_0', F')$ (AFD). Como construir A' a partir de A ?
 - a) $Q' = P(Q)$. Um elemento de Q' será representado por $[q_1q_2...q_i]$ onde $q_1, ..., q_i \in Q$.
 - b) $F' =$ conjunto formado pelos estados de Q' que possuem pelo menos um estado em F .
 - c) $q_0' = [q_0]$
 - d) $\delta'([q_1q_2...q_i], a) = [r_1r_2...r_j] \Leftrightarrow \delta(q_1, a) \cup \delta(q_2, a) \cup ... \cup \delta(q_i, a) = \{ r_1, r_2, ..., r_j \}$
- ♦ Exercício: Mostrar que $L(A') = L(A)$.
- ♦ Exemplo: Seja $A = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$ com δ dada por:



Vamos determinar o autômato determinístico A' equivalente a A .

Autômato Finito Não Determinístico

$A' = (Q, \{0,1\}, \delta', [q_0], F)$ tal que:

$$Q = \{ \emptyset, [q_0], [q_1], [q_0 q_1] \}$$

$$F = \{ [q_1], [q_0 q_1] \}$$

δ' dada por:

$$\delta'(\emptyset, 0) = \delta'(\emptyset, 1) = \emptyset$$

$$\delta(q_0, 0) = \{ q_0, q_1 \} \Rightarrow$$

$$\delta'([q_0], 0) = [q_0 q_1]$$

$$\delta(q_0, 1) = \{ q_1 \} \Rightarrow$$

$$\delta'([q_0], 1) = [q_1]$$

$$\delta(q_1, 0) = \emptyset \Rightarrow$$

$$\delta'([q_1], 0) = \emptyset$$

$$\delta(q_1, 1) = \{ q_0, q_1 \} \Rightarrow$$

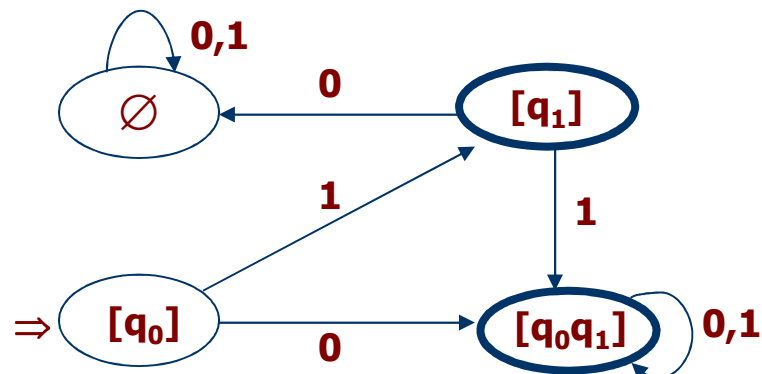
$$\delta'([q_1], 1) = [q_0 q_1]$$

e ainda:

$$\delta'([q_0 q_1], 0) = [q_0 q_1] \text{ (pois } \delta(q_0, 0) \cup \delta(q_1, 0) = \{ q_0, q_1 \} \text{)}$$

$$\delta'([q_0 q_1], 1) = [q_0 q_1] \text{ (pois } \delta(q_0, 1) \cup \delta(q_1, 1) = \{ q_0, q_1 \} \text{)}$$

Portanto, A' é o autômato:

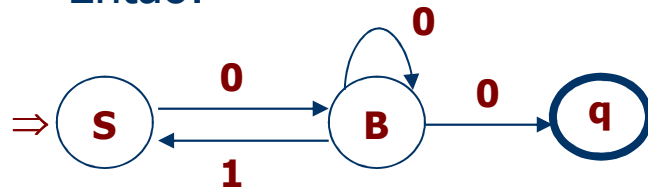


Autômato Finito Não Determinístico

- ♦ Teorema: Seja $G = (N, \Sigma, P, S)$, com $V = N \cup \Sigma$, uma gramática regular. Então existe autômato finito $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L(A) = L(G)$.
- ♦ Prova (Informal. Idéia da prova: fazer as transições do autômato simularem as derivações na gramática)

Construção de A (um AFND):

- a) $Q = N \cup \{q\}$, $q \notin N$
 - b) $q_0 = S$
 - c) $F = \{S, q\}$ se $S \rightarrow \Lambda \in P$; Caso contrário, $F = \{q\}$.
 - d) δ definida por:
 - 1) $q \in \delta(B, a)$ se $B \rightarrow a \in P$
 - 2) colocar em $\delta(B, a)$ os estados C tais que $B \rightarrow aC \in P$
 - 3) $\delta(q, a) = \emptyset$, $\forall a \in \Sigma$
- ♦ Exercício: Mostrar que $L(A) = L(G)$.
 - ♦ Exemplo: $G = (\{S, B\}, \{0, 1\}, P, S)$ com $P = \{S \rightarrow 0B, B \rightarrow 0B, B \rightarrow 1S, B \rightarrow 0\}$
Então:



Como descrever $L(A)$?

$L(A)$ é o conjunto das cadeias de $\{0, 1\}^*$ que começam com 0, terminam com 0 e que, caso contenha, cada 1 é seguido imediatamente por um 0, e ... Não há uma forma melhor de descrever $L(A)$?

Autômato Finito e Gramáticas Regulares

- ♦ Notação: Transição no autômato finito: $(q, ax) \mapsto (q', x) \Leftrightarrow \delta(q, a) = q'$
Com essa notação podemos escrever:

$$L(A) = \{ x \in \Sigma^* \mid (q_0, x) \mapsto^* (q, \Lambda), q \in F \}$$

- ♦ Exemplo: Para a gramática e autômato do exemplo anterior, podemos escrever:

$$S \Rightarrow 0B \Rightarrow 00B \Rightarrow 001S \Rightarrow 0010B \Rightarrow 00100$$

$$(S, 00100) \mapsto (B, 0100) \mapsto (B, 100) \mapsto (S, 00) \mapsto (B, 0) \mapsto (q, \Lambda)$$

- ♦ Teorema: Seja $A = (Q, \Sigma, \delta, q_0, F)$ um autômato finito. Então existe uma gramática regular G tal que $L(G) = L(A)$.
- ♦ Prova: Seja a gramática $G = (N, \Sigma, P, S)$ tal que:

1. $N = Q$

2. $S = q_0$

3. P construído por:

a) $q \rightarrow aq' \in P$ se $\delta(q, a) = q'$ $q, q' \in Q; a \in \Sigma$

b) $q \rightarrow a \in P$ se $\delta(q, a) \in F$ $a \in \Sigma \cup \{\Lambda\}$

Devemos mostrar que a gramática G assim construída é tal que $L(G) = L(A)$, ou seja, para $\forall x \in \Sigma^*$ devemos ter:

$$q_0 \Rightarrow^* x \Leftrightarrow (q_0, x) \mapsto^* (q, \Lambda), q \in F$$

Autômato Finito e Gramáticas Regulares

Inicialmente, vamos mostrar (por indução no comprimento de x) que para $\forall q \in Q$:

$$q \Rightarrow^* x \iff (q, x) \mapsto^* (q', \Lambda), q' \in F$$

Base: $i = 0$ (ou seja, $x = \Lambda$)

$$q \Rightarrow \Lambda \iff (q, \Lambda) \mapsto^0 (q, \Lambda), q \in F$$

pois por (3b) temos: $q \rightarrow \Lambda \in P$ se $\delta(q, \Lambda) \in F$

(note que o mesmo vale para $i = 1$, ou seja, $x = a \in \Sigma$)

Hipótese indutiva: a tese vale para $|x| = i, i \geq 0$, ou seja:

$$q \Rightarrow^i x \iff (q, x) \mapsto^i (q', \Lambda), q' \in F$$

Passo da indução: vamos mostrar que a tese vale para $|x| = i+1$.

Seja $x = aw$ tal que $|w| = i, a \in \Sigma$.

$$q \Rightarrow^{i+1} x \iff q \Rightarrow aq' \Rightarrow^i aw, \text{ ou seja: } q' \Rightarrow^i w \text{ e } \delta(q, a) = q' \text{ (pois } q \rightarrow aq' \in P)$$

Mas, pela hipótese indutiva:

$$q' \Rightarrow^i w \iff (q', w) \mapsto^i (q'', \Lambda), q'' \in F. \text{ Logo:}$$

$$q \Rightarrow^{i+1} x \iff (q, aw) \mapsto (q', w) \mapsto^i (q'', \Lambda), q'' \in F$$

$$\iff (q, x) \mapsto^{i+1} (q'', \Lambda), q'' \in F$$

Portanto:

$$\text{para } \forall q \in Q: \quad q \Rightarrow^* x \iff (q, x) \mapsto^* (q', \Lambda), q' \in F$$

Autômato Finito e Gramáticas Regulares

Como esse resultado vale para $\forall q \in Q$, podemos particularizar para q_0 :

$$q_0 \Rightarrow^* x \Leftrightarrow (q_0, x) \mapsto^* (q', \Lambda), q' \in F$$

$$x \in L(G) \Leftrightarrow x \in L(A)$$

Logo: $L(G) = L(A)$.

Expressões Regulares

Por que expressões regulares?

Uma forma simples de descrever a linguagem aceita por autômatos finidos (ou gerada por gramáticas regulares).

- ♦ Definição: Seja Σ um alfabeto. As expressões regulares sobre Σ e os conjuntos de cadeias que elas representam são definidos, recursivamente, como:
 - a) \emptyset é uma expressão regular e representa o conjunto vazio \emptyset .
 - b) Λ é uma expressão regular e representa o conjunto $\{ \Lambda \}$.
 - c) se $a \in \Sigma$ então a é uma expressão regular e representa o conjunto $\{ a \}$.
 - d) se r e s são expressões regulares representando, respectivamente, os conjuntos R e S , então $(r + s)$, (rs) , (r^*) são expressões regulares representando $R \cup S$, $RS = \{ rs \in \Sigma^* \mid r \in R \text{ e } s \in S \}$ e R^* , respectivamente.
- ♦ Para eliminar alguns parênteses nas expressões regulares, podemos assumir que a prioridade das operações é: (i) fecho, (ii) concatenação, (iii) soma.
- ♦ **Exemplo: $((0(1^*)) + 1)$ pode ser escrita como $01^* + 1$.**

Autômatos Finitos e Expressões Regulares

- ♦ Teorema: Seja r uma expressão regular e $L(r)$ a linguagem representada por r . Então existe um AFND A tal que $L(A) = L(r)$.
- ♦ Prova (por indução sobre o número de operadores na expressão r)
Base: r tem 0 operadores. Neste caso r pode ser \emptyset , Λ ou a ($a \in \Sigma$). Mas essas expressões representam conjuntos que são reconhecidos, respectivamente, por:



Hipótese indutiva: o teorema é válido para expressões regulares com menos que i operadores ($i \geq 1$).

Passo da indução: r tem i operadores. Neste caso podemos escrever que:

- a) $r = r_1 + r_2$, ou
- b) $r = r_1 r_2$, ou
- c) $r = r_1^*$

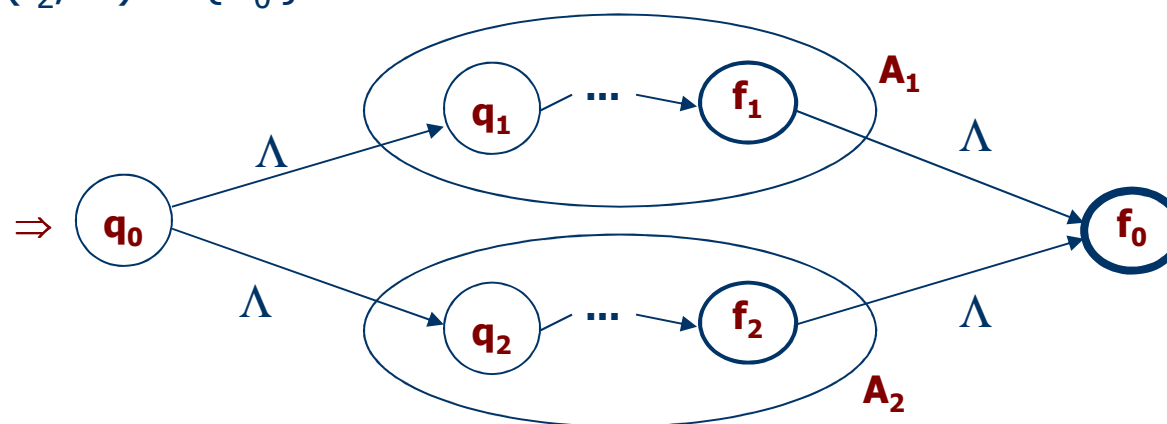
Autômatos Finitos e Expressões Regulares

(a) $r = r_1 + r_2$. Logo, r_1 e r_2 tem menos do que i operadores. Portanto, existem $A_1 = (Q_1, \Sigma, \delta_1, q_1, \{f_1\})$ e $A_2 = (Q_2, \Sigma, \delta_2, q_2, \{f_2\})$ tais que $L(A_1) = L(r_1)$ e $L(A_2) = L(r_2)$. Devemos construir A tal que $L(A) = L(r)$. Seja então:

$A = (\{Q_1 \cup Q_2 \cup \{q_0, f_0\}, \Sigma, \delta, q_0, \{f_0\})$ com δ definida por:

- 1) $\delta(q_0, \Lambda) = \{q_1, q_2\}$
- 2) $\delta(q, a) = \delta_1(q, a)$ se $q \in Q_1 - \{f_1\}$, $a \in \Sigma \cup \{\Lambda\}$
- 3) $\delta(q, a) = \delta_2(q, a)$ se $q \in Q_2 - \{f_2\}$, $a \in \Sigma \cup \{\Lambda\}$
- 4) $\delta(f_1, \Lambda) = \delta(f_2, \Lambda) = \{f_0\}$

Intuitivamente:



Então: $x \in L(r) \Leftrightarrow x \in L(r_1) \cup L(r_2) \Leftrightarrow (q_1, x) \mapsto^* (f_1, \Lambda)$ ou $(q_2, x) \mapsto^* (f_2, \Lambda)$
 $\Leftrightarrow (q_0, x) \mapsto (q_1, x) \mapsto^* (f_1, \Lambda) \mapsto (f_0, \Lambda)$ ou $(q_0, x) \mapsto (q_2, x) \mapsto^* (f_2, \Lambda) \mapsto (f_0, \Lambda)$
 $\Leftrightarrow x \in L(A)$. Logo: $L(r) = L(A)$.

- ♦ **Exercício:** Completar a prova para: (b) $r = r_1 r_2$ e (c) $r = r_1^*$

Autômatos Finitos e Expressões Regulares

- Exemplo: $r = 01^* + 1$. Vamos construir autômato A tal que $L(A) = L(r)$. Podemos escrever: $r = r_1 + r_2$ com $r_1 = 01^*$ e $r_2 = 1$



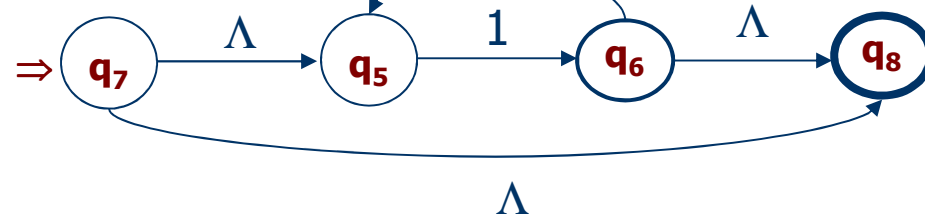
Seja $r_1 = r_3 r_4$ com $r_3 = 0$ e $r_4 = 1^*$.
Autômato para r_3 :



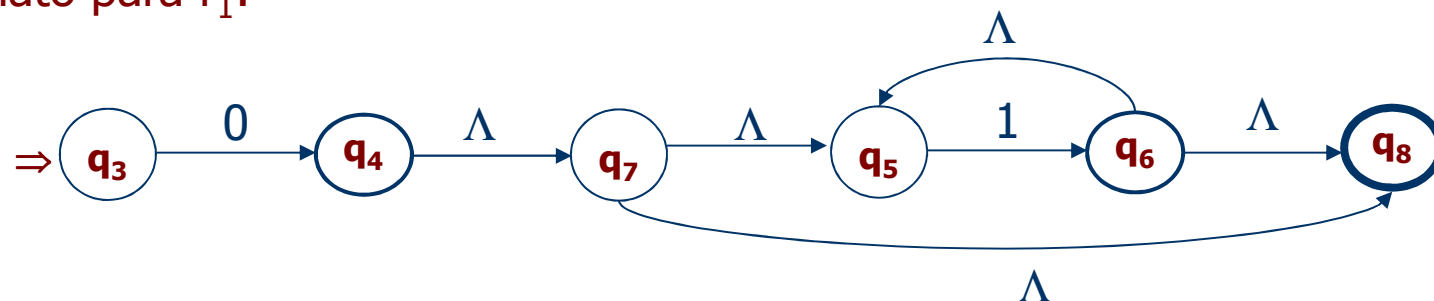
Seja $r_4 = r_5^*$ com $r_5 = 1$.
Autômato para r_5 :



Autômato para r_4 :

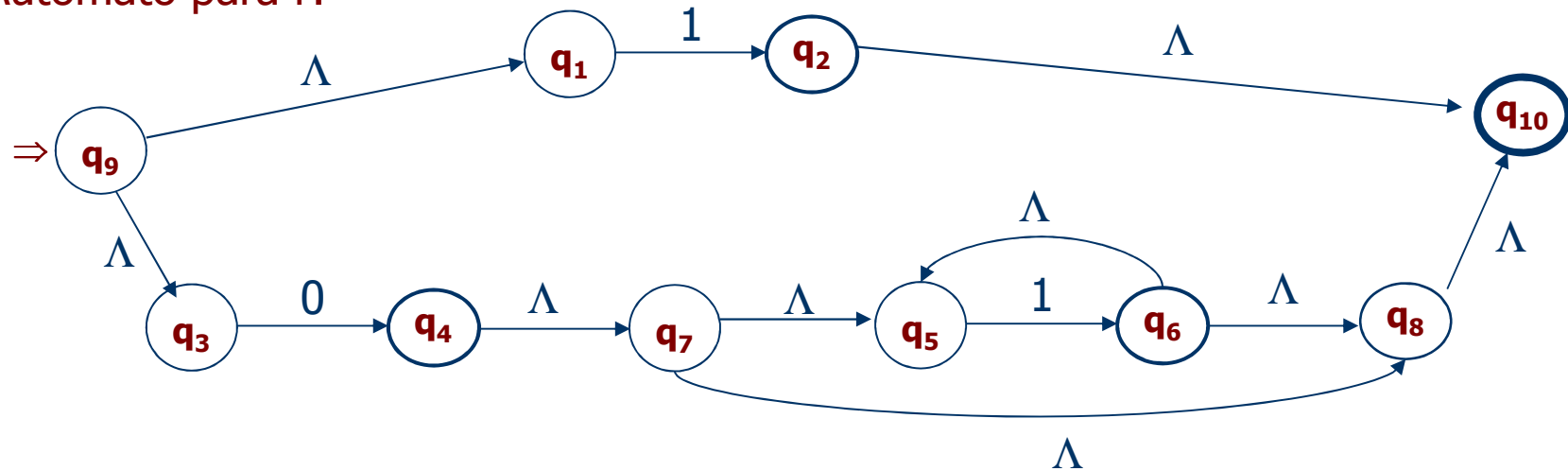


Autômato para r_1 :



Autômatos Finitos e Expressões Regulares

Autômato para r :



- ♦ Teorema: Seja $L = L(A)$ para algum autômato finito (determinístico) A . Então, $L = L(r)$ para alguma expressão regular r .
- ♦ Prova: Seja $A = (\{ q_1, \dots, q_n \}, \Sigma, \delta, q_1, F)$. Seja R_{ij}^m o conjunto das cadeias $x \in \Sigma^*$ tais que $\delta(q_i, x) = q_j$ e se $\delta(q_i, y) = q_k$ para y prefixo de x ($y \neq \Lambda$, $y \neq x$) então $k \leq m$. Intuitivamente, R_{ij}^m é o conjunto das cadeias que fazem o autômato ir do estado q_i para o estado q_j sem passar (isto é, entrar e sair) por qualquer estado de número maior do que m (notar que i, j podem ser maiores do que m). Observe que, como não existe estado de número maior do que n , R_{ij}^n é o conjunto de todas as cadeias que levam o autômato do estado q_i para o estado q_j .
Podemos definir R_{ij}^m , recursivamente, como:

Autômatos Finitos e Expressões Regulares

$$\begin{aligned} R_{ij}^0 &= \{ a \in \Sigma \mid \delta(q_i, a) = q_j \} && \text{se } i \neq j \\ R_{ij}^0 &= \{ a \in \Sigma \mid \delta(q_i, a) = q_j \} \cup \{ \Lambda \} && \text{se } i = j \end{aligned}$$

$$R_{ij}^m = R_{ij}^{m-1} \cup R_{im}^{m-1} (R_{mm}^{m-1})^* R_{mj}^{m-1}$$

Devemos mostrar que para todo i, j, m existe uma expressão regular r_{ij}^m que representa R_{ij}^m . Vamos mostrar por indução em m .

Base: $m = 0$. R_{ij}^0 é um conjunto de símbolos x tais que $x \in \Sigma$ ou $x = \Lambda$, ou então $R_{ij}^0 = \emptyset$. Se $R_{ij}^0 = \{ a_1, \dots, a_p \}$, $a_i \in \Sigma \cup \{ \Lambda \}$, então $r_{ij}^0 = a_1 + \dots + a_p$. Se $R_{ij}^0 = \emptyset$ então $r_{ij}^0 = \emptyset$.

Passo: a definição recursiva de R_{ij}^m envolve apenas operadores de expressões regulares (união, concatenação e fecho). Por hipótese indutiva, para todo i, j existe uma expressão regular r_{ij}^{m-1} tal que $R_{ij}^{m-1} = L(r_{ij}^{m-1})$. Logo:

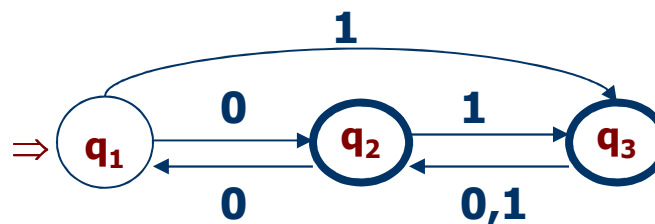
$$r_{ij}^m = r_{ij}^{m-1} + (r_{im}^{m-1})(r_{mm}^{m-1})^*(r_{mj}^{m-1})$$

Para completar a prova, devemos observar que: $L(A) = \cup R_{1j}^n$, $q_j \in F$, e portanto, $L(A)$ pode ser representado por:

$$r_{1j_1}^n + \dots + r_{1j_p}^n \quad \text{tal que} \quad F = \{ q_{j_1}, \dots, q_{j_p} \}$$

Autômatos Finitos e Expressões Regulares

- Exemplo: Seja o autômato



Nesse caso, $L(A) = R_{12}^3 \cup R_{13}^3 = r_{12}^3 + r_{13}^3$

	k = 0	k = 1	k = 2	k = 3
r_{11}^k	Λ	$r_{11}^1 = r_{11}^0 + (r_{11}^0)(r_{11}^0)^*(r_{11}^0) = \Lambda$		
r_{12}^k	0	$r_{12}^1 = r_{12}^0 + (r_{11}^0)(r_{11}^0)^*(r_{12}^0) = 0 + 0 = 0$		
r_{13}^k	1			
r_{21}^k	0			
r_{22}^k	Λ			
r_{23}^k	1			
r_{31}^k	\emptyset			
r_{32}^k	0 + 1			
r_{33}^k	Λ			

Exercício: Completar essa tabela e determinar $L(A)$!

“Pumping Lemma” para Conjuntos Regulares

- ♦ O teorema a seguir estabelece uma condição necessária sobre as cadeias de um conjunto regular. O teorema é um resultado importante para provar que certos conjuntos não são regulares.
- ♦ Teorema: Seja $A = (Q, \Sigma, \delta, q_0, F)$ um autômato finito com n estados. Seja $w \in L(A)$ tal que $|w| \geq p$. Se $p \geq n$ então existem $x, y, z \in \Sigma^*$ tais que: $w = xyz$, $y \neq \Lambda$, e para todo $k \geq 0$, $xy^kz \in L(A)$.
- ♦ Prova: Sejam: $p = n$, $w \in L(A)$, $|w| \geq n$. Portanto, durante o processamento de w o autômato utiliza pelo menos $(n+1)$ estados. Como $\#Q = n$, deve haver repetição de pelo menos um estado. Sejam então x, y e $z \in \Sigma^*$, com $y \neq \Lambda$, tais que: $(q_0, xyz) \mapsto^* (q, yz) \mapsto^+ (q, z) \mapsto^* (q', \Lambda)$, $q' \in F$. Logo, se $(q, yz) \mapsto^+ (q, z)$ então $(q, y^kz) \mapsto^+ (q, y^{k-1}z) \mapsto^+ \dots \mapsto^+ (q, z)$ e portanto, $xy^kz \in L(A)$, $k \geq 0$.
- ♦ Exemplo: $L = \{ 0^n 1^n \mid n \geq 1 \}$ não é regular.
Admitindo que L seja regular, existe um autômato finito $A = (Q, \Sigma, \delta, q_0, F)$ com p estados tal que $L = L(A)$. Como o número p de estados do autômato é finito e como em L existem cadeias arbitrariamente grandes, é possível encontrar um n tal que $|0^n 1^n| \geq p$. Neste caso, pelo “pumping lemma”, existem $x, y, z \in \{0,1\}^*$ tais que: $0^n 1^n = xyz$, $y \neq \Lambda$ e $xy^kz \in L$ ($k \geq 0$). Neste caso y pode ser de apenas três formas:
a) $y \in L(0^+)$ e, neste caso, por exemplo, $xz \notin L$

Propriedades de Conjuntos Regulares

- b) $y \in L(1^+)$ e, neste caso, por exemplo, $xz \notin L$
c) $y \in L(0^+1^+)$ e, neste caso, por exemplo, $xy^2z \notin L$

Portanto, a hipótese de L ser regular contradiz o “pumping lemma”.

Logo, $L = \{ 0^n 1^n \mid n \geq 1 \}$ não é regular.

- ♦ Teorema: A classe de linguagens regulares é fechada sob complementação, ou seja, se L é uma linguagem regular sobre Σ^* então $\Sigma^* - L$ é linguagem regular.
- ♦ Prova: Como L é regular, existe $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L = L(A)$.
Seja $A' = (Q, \Sigma, \delta, q_0, Q - F)$. Neste caso:
 $x \in \Sigma^* - L \Leftrightarrow x \notin L \Leftrightarrow \delta(q_0, x) \notin F \Leftrightarrow \delta(q_0, x) \in Q - F \Leftrightarrow x \in L(A')$.
Logo, $L(A') = \Sigma^* - L$ e, portanto, $\Sigma^* - L$ é regular.
- ♦ Teorema: A classe de linguagens regulares é fechada sob união, ou seja, se X e Y são conjuntos regulares, então $X \cup Y$ é conjunto regular.
- ♦ Prova: Sejam $A = (Q_A, \Sigma, \delta_A, q_{0A}, F_A)$ e $B = (Q_B, \Sigma, \delta_B, q_{0B}, F_B)$ tais que $X = L(A)$ e $Y = L(B)$. Seja $q' \notin Q_A \cup Q_B$ e $C = (Q_A \cup Q_B \cup \{ q' \}, \Sigma, \delta_C, q', F_C)$ (AFND) tal que:
$$F_C = F_A \cup F_B \quad \text{se } \Lambda \notin L(A) \cup L(B)$$
$$F_C = F_A \cup F_B \cup \{ q' \} \quad \text{se } \Lambda \in L(A) \cup L(B)$$

Propriedades de Conjuntos Regulares

e δ_C definida por:

$$\delta_C(q', a) = \{ \delta_A(q_{0A}, a) \} \cup \{ \delta_B(q_{0B}, a) \}, a \in \Sigma \cup \{ \Lambda \}$$

$$\delta_C(q, a) = \delta_A(q, a), q \in Q_A, a \in \Sigma \cup \{ \Lambda \}$$

$$\delta_C(q, a) = \delta_B(q, a), q \in Q_B, a \in \Sigma \cup \{ \Lambda \}$$

Seja $x \in X \cup Y$. Dois casos podem ocorrer:

a) $x = \Lambda$. Logo, $\Lambda \in L(A)$ ou $\Lambda \in L(B)$. Portanto, $\delta_C(q', \Lambda) = q' \in F_C$. Logo: $\Lambda \in L(C)$.

b) $x \neq \Lambda$. Seja $x = aw$, $a \in \Sigma$, $w \in \Sigma^*$. Então:

$$aw \in X \cup Y \Leftrightarrow aw \in X \text{ ou } aw \in Y \Leftrightarrow$$

$$\Leftrightarrow (q_{0A}, aw) \mapsto (q_{1A}, w) \mapsto^* (q_{fA}, \Lambda), q_{fA} \in F_A \text{ ou}$$

$$(q_{0B}, aw) \mapsto (q_{1B}, w) \mapsto^* (q_{fB}, \Lambda), q_{fB} \in F_B$$

$$\Leftrightarrow (q', aw) \mapsto (q_{1A}, w) \mapsto^* (q_{fA}, \Lambda) \text{ ou } (q', aw) \mapsto (q_{1B}, w) \mapsto^* (q_{fB}, \Lambda)$$

$$\Leftrightarrow \delta_C(q', aw) \in F_A \cup F_B \Leftrightarrow aw \in L(C).$$

Logo: $L(C) = X \cup Y$ e, portanto, $X \cup Y$ é regular.

- ♦ Teorema: A classe de linguagens regulares é fechada sob interseção, ou seja, se X e Y são conjuntos regulares, então $X \cap Y$ é conjunto regular.
- ♦ **Prova: Exercício!**

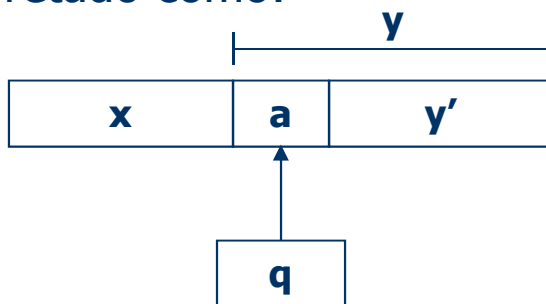
Autômatos Finitos Bilaterais

- Definição: Um autômato finito bilateral (AFB) é uma quintupla $A = (Q, \Sigma, \delta, q_0, F)$ onde Q, Σ, q_0 e F são definidos como no caso do autômato finito e δ é uma função da forma:

$$\delta : Q \times \Sigma \rightarrow \{-1, 0, 1\} \times Q$$

$\delta(q, a) = (d, q') \Leftrightarrow$ o autômato estando no estado q e lendo o símbolo a , move sua cabeça uma posição para a esquerda, se $d = -1$, permanece estacionário, se $d = 0$, ou move sua cabeça uma posição para a direita, se $d = 1$, e vai para o estado q' .

- Definição: Uma configuração de um AFB $A = (Q, \Sigma, \delta, q_0, F)$ é um elemento de $\Sigma^*Q\Sigma^*$ (xqy), interpretado como:



- Definição: $a_1 \dots a_{i-1} q a_i \dots a_n \mapsto a_1 \dots q' a_{i+d} \dots a_n \Leftrightarrow \delta(q, a_i) = (d, q')$
- Definição: A linguagem reconhecida por um AFB $A = (Q, \Sigma, \delta, q_0, F)$ é definida como: $L(A) = \{ x \in \Sigma^* \mid q_0 x \mapsto^* xq, q \in F \}$.

Autômatos Finitos Bilaterais

- Um autômato finito bilateral rejeita cadeias de uma das seguintes maneiras:
 - saindo à esquerda
 - entrando em "loop"
 - saindo à direita para um estado não final
- Exemplo: $A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$ com δ dada por:

δ	0	1
q_0	(1, q_1)	(-1, q_1)
q_1	(-1, q_0)	(0, q_2)
q_2	(-1, q_1)	(1, q_2)

Para esse autômato, temos:

a) $q_0 0 1 1 \mapsto 0 q_1 1 1 \mapsto 0 q_2 1 1 \mapsto 0 1 q_2 1 \mapsto 0 1 1 q_2$

e, como $q_2 \in F$, a cadeia $0 1 1 \in L(A)$

b) $q_0 0 0 \mapsto 0 q_1 0 \mapsto q_0 0 0 \mapsto 0 q_1 0 \mapsto \dots$ "loop"

portanto: $0 0 \notin L(A)$.

- Teorema: Para todo AFB $A = (Q, \Sigma, \delta, q_0, F)$, $L(A)$ é um conjunto regular.
- Prova (informal):
Para todo $x \in \Sigma^*$, definir a função: $\alpha_x : Q \cup \{0\} \rightarrow Q \cup \{0\}$, $0 \notin Q$ da seguinte maneira:
 - se $x = x'a$ ($a \in \Sigma$, $x' \in \Sigma^*$) e $x'qa \mapsto^* x'aq'$ então $\alpha_x(q) = q'$. Caso contrário, $\alpha_x(q) = 0$.
 - se $q_0x \mapsto^* xq'$ então $\alpha_x(0) = q'$. Caso contrário, $\alpha_x(0) = 0$.

Autômatos Finitos Bilaterais

Utilizando a relação de Nerode:

$$x R_{L(A)} y \Leftrightarrow (\forall z \in \Sigma^*) (xz \in L(A) \Leftrightarrow yz \in L(A))$$

podemos provar que $\alpha_x = \alpha_y \Rightarrow x R_{L(A)} y$. Deste resultado segue que:
 $i(R_{L(A)}) \leq \#\{\alpha_x \mid x \in \Sigma^*\}$. Mas, se $\#Q = n$, então o número de funções α_x diferentes é, no máximo, $(n+1)^{n+1}$. Portanto, $i(R_{L(A)})$ é finito e $L(A)$ é regular.

A idéia da prova desse teorema é de:

SHEPHERDSON, J.C. "The reduction of two-way automata to one-way automata", IBM Journal of Research and Development, 3, 1959.

Para uma prova formal ver também:

HARRISON, M.A. "Introduction to formal language theory", p. 65-70.

Problemas de Decisão

- ♦ Definição: Um problema de decisão tem solução se e somente se existe um algoritmo que resolve o problema para todas as possíveis condições. Se tal algoritmo existe, o problema é decidível; caso contrário, o problema é indecidível.

Problemas de Decisão

- ♦ Exemplos:
- ♦ O problema: “A linguagem reconhecida por um autômato A é vazia?” é decidível.
Idéia da prova: $A = (Q, \Sigma, \delta, q_0, F)$. Verificar se $\exists q \in Q$ tal que $q \in F$ e q é acessível.
- ♦ O problema: “Sejam X e Y conjuntos regulares. $X \subseteq Y$ ou $X = Y$?” é decidível.
Idéia da prova: Sejam $X = L(A)$ e $Y = L(B)$ para dois autômatos finitos A e B . Então existe um algoritmo para verificar se A e B são equivalentes:
 $A \equiv B$ se para todo estado q_i de A existe um estado q_j de B tal que $q_i \equiv q_j$ e reciprocamente. Basta então construir o autômato que reconhece $X \cup Y$ e aplicar o algoritmo de minimização de estados.
Então: $X = Y$ se $A \equiv B$.
Por outro lado, se $X \subseteq Y$ então $X \cap Y = X$. Logo, para saber se $X \subseteq Y$, basta construir o autômato que reconhece $X \cap Y$ e verificar se ele é equivalente a A .
(uma outra idéia para resolver esse problema de decisão é mostrada em: MANNA, Z. “Mathematical theory of computation”).
- ♦ O problema: “Seja X um conjunto regular. X é finito?” é decidível.
- ♦ Prova: Exercício! (sugestão: “pumping lemma”).