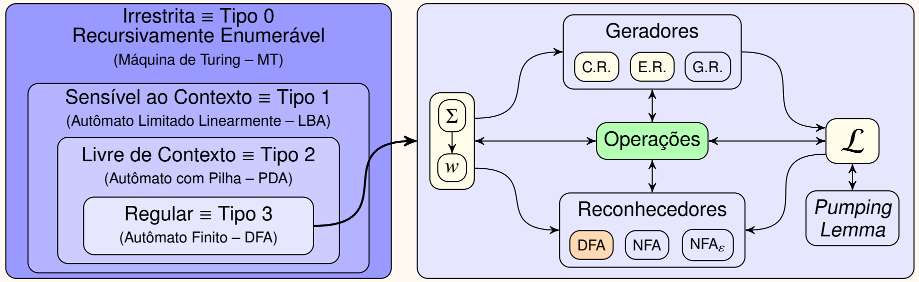
Linguagens Formais e Autômatos

Humberto Longo

Instituto de Informática
Universidade Federal de Goiás

Bacharelado em Ciência da Computação, 2021/1

Roteiro



Complemento de um DFA

Teorema 1.28

- *Se $M = \langle \Sigma, S, s_0, \delta, F \rangle$ é um DFA, então $M' = \langle \Sigma, S, s_0, \delta, S - F \rangle$ é um DFA tal que $\mathcal{L}(M') = \Sigma^* - \mathcal{L}(M)$.*

Demonstração.

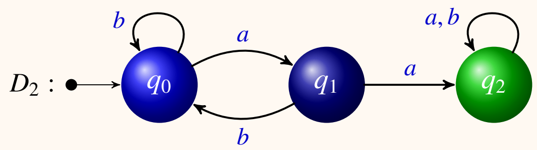
- Seja $w \in \Sigma^*$ e $\overline{\delta}$ a função de transição estendida construída a partir de δ.
- Para cada $w \in \mathcal{L}(M)$, $\overline{\delta}(s_0, w) \in F$.
- Portanto, $w \notin \mathcal{L}(M')$.
- Se $w \notin \mathcal{L}(M)$, então $\overline{\delta}(s_0, w) \in S - F$ e $w \in \mathcal{L}(M')$.

□

Complemento de um DFA

Exemplo 1.29

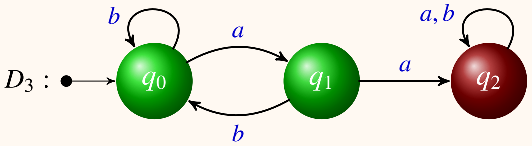
- DFA D_2 que reconhece $\mathcal{L}_2 = \{w \in \{a, b\}^* \mid w$ contém a subcadeia $aa\}$.



Complemento de um DFA

Exemplo 1.29

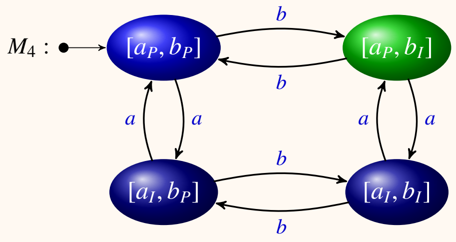
- DFA D_3 que reconhece $\mathcal{L}_3 = \{w \in \{a,b\}^* \mid w$ não contém a subcadeia $aa\}$.



Complemento de um DFA

Exemplo 1.30

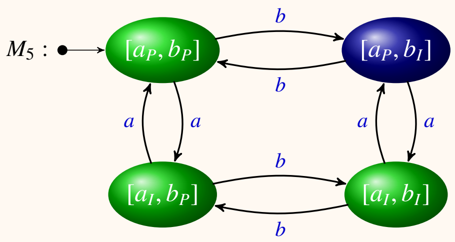
- $\mathcal{L}(M_4) = \{w \in \{a,b\}^+ \mid w$ contém nr. par de $a's$ e nr. ímpar de b's$\}$.



Complemento de um DFA

Exemplo 1.30

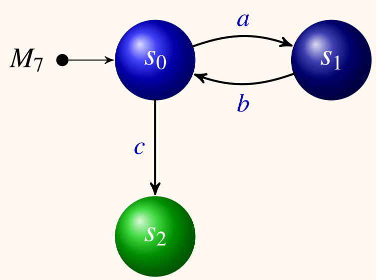
- $\mathcal{L}(M_5) = \{w \in \{a,b\}^+ \mid w$ não contém nr. par de $a's$ e nr. ímpar de b's$\}$.
- $\mathcal{L}(M_5) = \{a,b\}^* - \mathcal{L}(M_4)$.



Complemento de um DFA

Exemplo 1.31

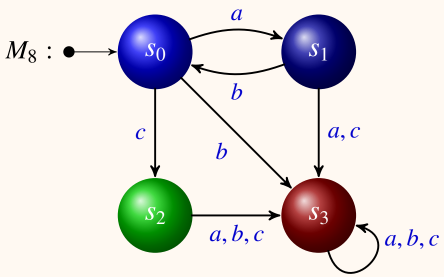
- $\mathcal{L}(M_7) = (ab)^*c$.



Complemento de um DFA

Exemplo 1.31

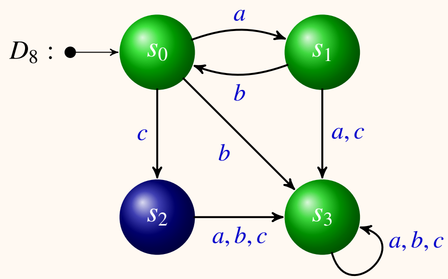
- $\mathcal{L}(M_7) = (ab)^*c$.



Complemento de um DFA

Exemplo 1.31

- $\mathcal{L} = \Sigma^* - \mathcal{L}(M_7)$.



Operações com autômatos

Produto $\otimes$ de autômatos

- Sejam os DFA's $D_1 = \langle \Sigma_1, S, s_0, \delta_1, F_1 \rangle$ e $D_2 = \langle \Sigma_2, Q, q_0, \delta_2, F_2 \rangle$.
- O produto de D_1 e D_2 é o autômato $D_1 \otimes D_2 = \langle \Sigma_1 \cap \Sigma_2, S.Q, s_0.q_0, \delta, F_1.F_2 \rangle$:
  - $S.Q = \{s_i.q_j \mid (s_i, q_j) \in S \times Q\}$,
  - $F_1.F_2 = \{s_i.q_j \mid (s_i, q_j) \in F_1 \times F_2\}$,
  - Para $s_i \in S$, $q_j \in Q$ e $a \in \Sigma_1 \cap \Sigma_2$, define-se

$$\delta(s_i.q_j, a) = \begin{cases} \delta_1(s_i, a).\delta_2(q_j, a), & \text{se } \delta_1(s_i, a) \text{ e } \delta_2(q_j, a) \text{ estão definidos;} \\ \emptyset, & \text{se } \delta_1(s_i, a) \text{ ou } \delta_2(q_j, a) \text{ não está definido.} \end{cases}$$

Produto de autômatos

Lema 1.32

- *Dados os DFA's $D_1 = \langle \Sigma_1, S, s_0, \delta_1, F_1 \rangle$, $D_2 = \langle \Sigma_2, Q, q_0, \delta_2, F_2 \rangle$ e $D_1 \otimes D_2 := \langle \Sigma_1 \cap \Sigma_2, S.Q, s_0.q_0, \delta, F_1.F_2 \rangle$, então $\mathcal{L}(D_1 \otimes D_2) = \mathcal{L}(D_1) \cap \mathcal{L}(D_2)$.*

Demonstração.

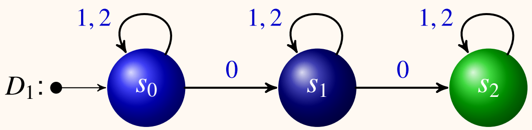
$$\begin{aligned} w \in \mathcal{L}(D_1 \otimes D_2) & \text{ se e somente se } \overline{\delta}(s_0.q_0, w) \in F_1.F_2, \\ & \text{se e somente se } \overline{\delta_1}(s_0, w).\overline{\delta_2}(q_0, w) \in F_1.F_2, \\ & \text{se e somente se } \overline{\delta_1}(s_0, w) \in F_1 \text{ e } \overline{\delta_2}(q_0, w) \in F_2, \\ & \text{se e somente se } w \in \mathcal{L}(D_1) \text{ e } w \in \mathcal{L}(D_2). \end{aligned}$$

□

Produto ⊗ de autômatos

Exemplo 1.33

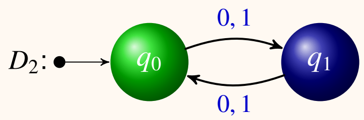
- DFA D_1 que reconhece $\mathcal{L}_1 = \{w \in \{0,1,2\}^* \mid w$ contém exatamente dois 0's$\}$:



Produto ⊗ de autômatos

Exemplo 1.33

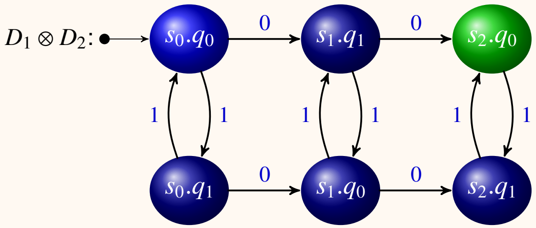
- DFA D_2 que reconhece $\mathcal{L}_2 = \{w \in \{0,1\}^* \mid w$ possui comprimento par$\}$:



Produto ⊗ de autômatos

Exemplo 1.33

- $\mathcal{L}_3 = \{w \in \{0,1\}^* \mid w$ contém exatamente dois 0's e possui comprimento par$\}$.
- $\mathcal{L}_3 = D_1 \otimes D_2 = \mathcal{L}_1 \cap \mathcal{L}_2$.
- DFA $D_3 = D_1 \otimes D_2$ que reconhece $\mathcal{L}_3$:



Operações com autômatos

Produto ⊕ de autômatos

- Sejam os DFA's $D_1 = \langle \Sigma_1, S, s_0, \delta_1, F_1 \rangle$ e $D_2 = \langle \Sigma_2, Q, q_0, \delta_2, F_2 \rangle$.
- $D_1 \oplus D_2 = \langle \Sigma_1 \cup \Sigma_2, S.Q, s_0.q_0, \delta, F_1.F_2 \rangle$:
  - $S.Q = \{s_i.q_j \mid (s_i, q_j) \in S \times Q\}$,
  - $F_1.F_2 = \{s_i.q_j \mid (s_i, q_j) \in S \times F_2$ ou $(s_i, q_j) \in F_1 \times Q\}$,
  - Para $s_i \in S$, $q_j \in Q$ e $a \in \Sigma_1 \cup \Sigma_2$, define-se

$$\delta(s_i.q_j, a) = \begin{cases} \delta_1(s_i, a).\delta_2(q_j, a), & \text{se } \delta_1(s_i,a) \text{ e } \delta_2(q_j,a) \text{ estão definidos;} \\ \varnothing, & \text{se } \delta_1(s_i,a) \text{ ou } \delta_2(q_j,a) \text{ não está definido.} \end{cases}$$

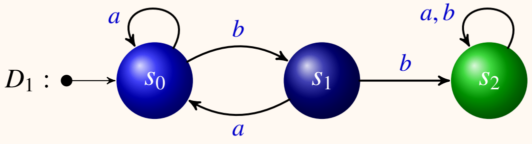
Lema 1.34

- *Dados os DFA's* $D_1 = \langle \Sigma_1, S, s_0, \delta_1, F_1 \rangle$, $D_2 = \langle \Sigma_2, Q, q_0, \delta_2, F_2 \rangle$ *e* $D_1 \oplus D_2 := \langle \Sigma_1 \cup \Sigma_2, S.Q, s_0.q_0, \delta, F_1.F_2 \rangle$, *então* $\mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.

Produto ⊕ de autômatos

Exemplo 1.35

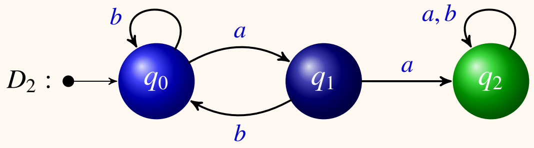
- DFA D_1 que reconhece
  $\mathcal{L}_1 = \{w \in \{a, b\}^* \mid w$ possui pelo menos dois b's consecutivos$\}$:



Produto ⊕ de autômatos

Exemplo 1.35

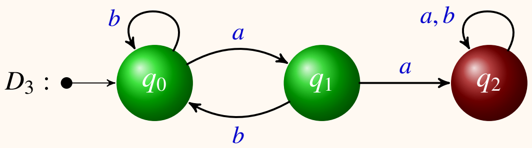
- DFA D_2 que reconhece
  $\mathcal{L}_2 = \{w \in \{a, b\}^* \mid w$ contém a subcadeia $aa\}$.



Produto ⊕ de autômatos

Exemplo 1.35

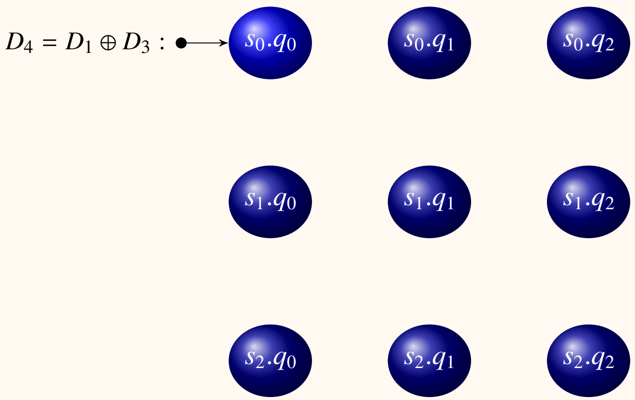
- DFA D_3 que reconhece
  $\mathcal{L}_3 = \{w \in \{a, b\}^* \mid w$ não contém a subcadeia $aa\}$.



Produto ⊕ de autômatos

Exemplo 1.35

- $\mathcal{L}(D_4) = \{w \in \{a, b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.

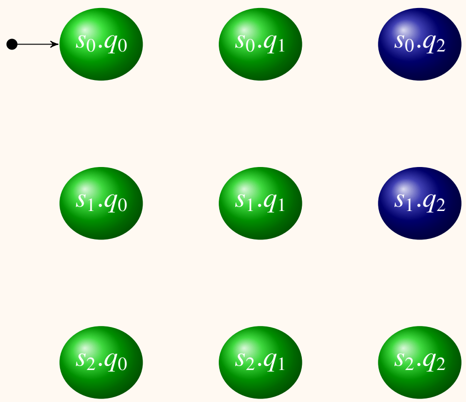


Produto ⊕ de autômatos

Exemplo 1.35

- $\mathcal{L}(D_4) = \{w \in \{a,b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.

$D_4 = D_1 \oplus D_3$:

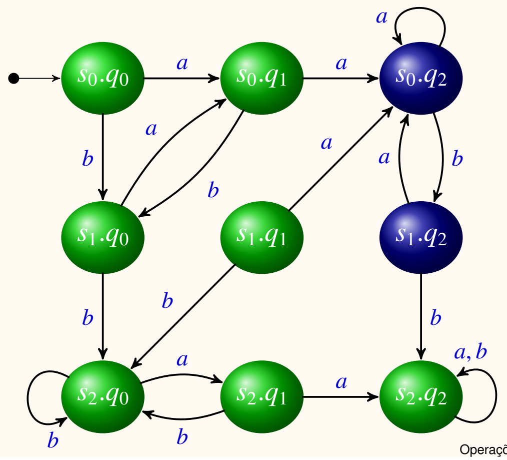


Produto ⊕ de autômatos

Exemplo 1.35

- $\mathcal{L}(D_4) = \{w \in \{a,b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.

$D_4 = D_1 \oplus D_3$:

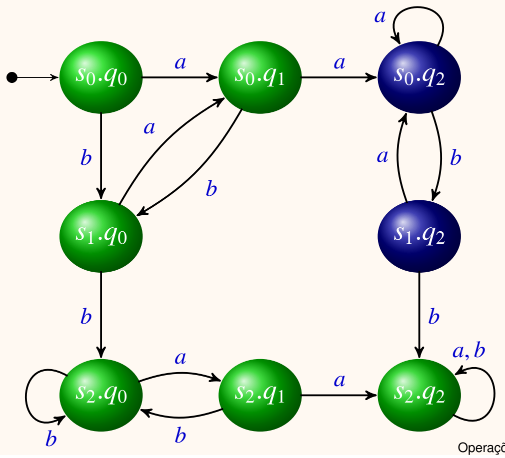


Produto ⊕ de autômatos

Exemplo 1.35

- $\mathcal{L}(D_4) = \{w \in \{a,b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.

$D_4 = D_1 \oplus D_3$:

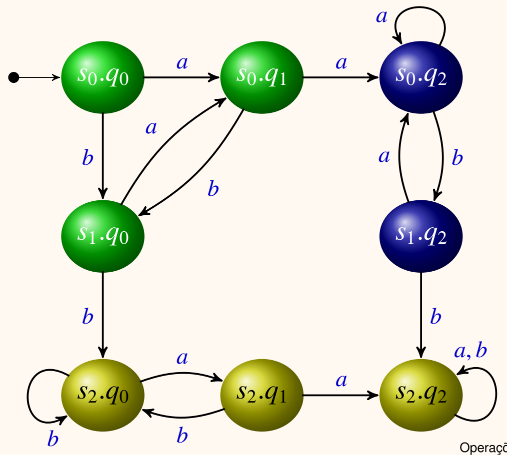


Produto ⊕ de autômatos

Exemplo 1.35

- $\mathcal{L}(D_4) = \{w \in \{a,b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.

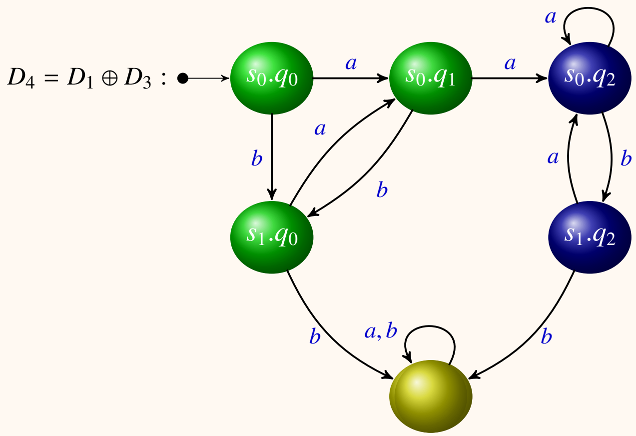
$D_4 = D_1 \oplus D_3$:



Produto ⊕ de autômatos

Exemplo 1.35

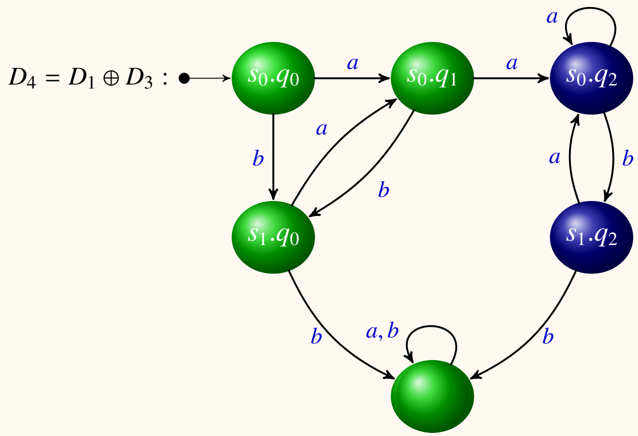
- $\mathcal{L}(D_4) = \{w \in \{a,b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.



Produto ⊕ de autômatos

Exemplo 1.35

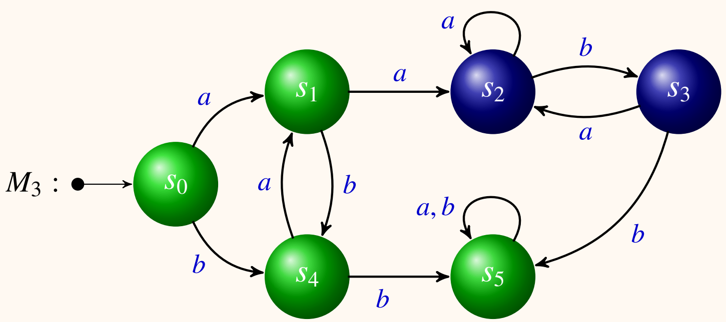
- $\mathcal{L}(D_4) = \{w \in \{a,b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1 \oplus D_2) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.



Produto ⊕ de autômatos

Exemplo 1.35

- $\mathcal{L}(D_4) = \{w \in \{a,b\}^* \mid w$ contém bb ou não contém $aa\}$.
- $\mathcal{L}(D_4) = \mathcal{L}(D_1) \cup \mathcal{L}(D_2)$.
- M_3 é outra versão minimizada do autômato D_4:



Minimização de DFA's – I

Definição 1.36

- Dado um DFA $M = \langle \Sigma, S, s_0, \delta, F \rangle$, é possível construir um DFA $M' = \langle \Sigma, S', s_0', \delta', F' \rangle$, a partir de M e da relação de não equivalência de estados (definida no teorema 1.38), onde:
  - S' é o conjunto das classes de equivalência constituídas de estados equivalentes de M;
  - $[s_0] \in S'$ se estado inicial $s_0 \in S$;
  - $[s_i] \in F'$ se $s_i \in F$;
  - $\delta'([s_i], a) = [\delta(s_i, a)]$.

Minimização de DFA's – I

Observações

- Se M' tem estados inalcançáveis a partir de $[s_0]$, estes estados e todos os arcos associados são eliminados.
- $\mathcal{L}(M') = \{u \mid \overline{\delta}'([s_0], u) = [\overline{\delta}(s_i, \varepsilon)]$ com $s_i \in F\}$.
- $\mathcal{L}(M) = \mathcal{L}(M')$.

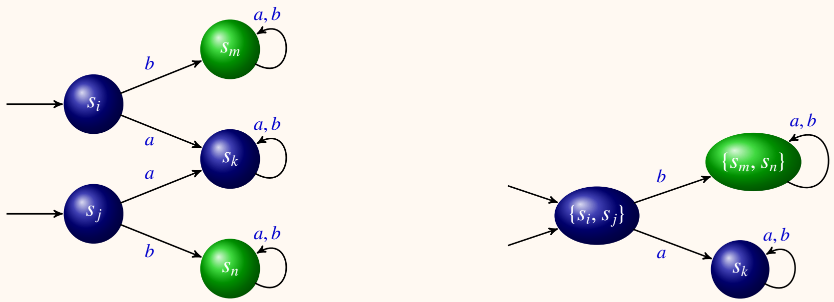
Minimização de DFA's – I

Definição 1.37 (Estados equivalentes)

- Seja o DFA $M = \langle \Sigma, S, s_0, \delta, F \rangle$. Dois estados $s_i, s_j \in S$ são equivalentes se $\overline{\delta}(s_i, u) \in F$ se, e somente se, $\overline{\delta}(s_j, u) \in F$, para todo $u \in \Sigma^*$.

Minimização de DFA's – I

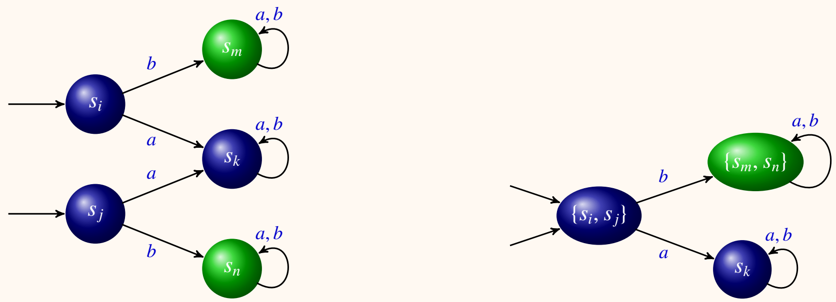
Estados equivalentes



- Estados s_i e s_j são equivalentes:
  - processamento de qualquer cadeia começando com b, a partir de s_i ou s_j, termina em um estado final; e
  - processamento de qualquer cadeia começando com a, a partir de s_i ou s_j, termina em um estado não final.

Minimização de DFA's – I

Estados equivalentes



- Estados s_m e s_n são equivalentes.
  - Qualquer processamento nestes estados termina em um estado final.

Minimização de DFA's – I

Identificação de estados equivalentes

- Para cada par de estados s_i e s_j ($i < j$), calcular os valores $NEq[i, j]$ e $Dep[i, j]$:
  - $NEq[i, j] = \begin{cases} 1, & \text{se estados } s_i \text{ e } s_j \text{ são não equivalentes;} \\ 0, & \text{caso contrário.} \end{cases}$
  - $(i, j) \in Dep[m, n] \Rightarrow$ não equivalência do par de estados (s_i, s_j) depende da não equivalência do par (s_m, s_n).

Minimização de DFA's – I

Identificação de estados equivalentes

$s_i \xrightarrow{a} s_m \qquad s_j \xrightarrow{a} s_n$

- Quando examinados os estados s_i e s_j:
  1. Se s_m e s_n já estão marcados como não equivalentes, então $NEq[i, j]$ recebe 1.
  2. Se s_m e s_n ainda não estão marcados como não equivalentes, então uma posterior determinação de que s_m e s_n são não equivalentes fornece a resposta para s_i e s_j.
     - $(i, j) \in Dep[m, n]$ indica que a não equivalência de s_m e s_n é suficiente para determinar a não equivalência de s_i e s_j.

Estados equivalentes

Algoritmo 1: EstadosEquivalentes(M)

`// Determina os estados equivalentes de um DFA` M

Entrada: DFA $M = \langle \Sigma, S, s_0, \delta, F \rangle$.
Saída: Matrizes NEq e Dep.

para todo $s_i, s_j \in S, i < j$, **faça**
    se $((s_i \in F$ **e** $s_j \notin F)$ **ou** $(s_i \notin F$ **e** $s_j \in F))$ **então** $NEq[i, j] \leftarrow 1$;
    senão $NEq[i, j] \leftarrow 0$;
    $Dep[i, j] \leftarrow \emptyset$;
para todo $((a \in \Sigma)$ **e** $(NEq[i, j] = 0, i < j))$ **faça**
    $s_m \leftarrow \delta(s_i, a)$;
    $s_n \leftarrow \delta(s_j, a)$;
    se $((NEq[m, n] = 1)$ **ou** $(NEq[n, m] = 1))$ **então** DIST(i, j);
    senão
        se $((m < n)$ **e** $((i, j) \neq (m, n)))$ **então**
          $Dep[m, n] \leftarrow Dep[m, n] \cup \{(i, j)\}$;
        senão
            se $((m > n)$ **e** $((i, j) \neq (n, m)))$ **então**
              $Dep[n, m] \leftarrow Dep[n, m] \cup \{(i, j)\}$;

Estados equivalentes

Algoritmo 2: DIST(i, j)

Entrada: Índices i e j.

$NEq[i, j] \leftarrow 1$;
para todo $(k, \ell) \in Dep[i, j]$ **faça**
    DIST(k, ℓ);

Estados equivalentes

Teorema 1.38

- *Os estados s_i e s_j são não equivalentes se, e somente se, $NEq[i,j] = 1$ ao término do Algoritmo 1.*

Demonstração.

1. $NEq[i,j] = 1 \Rightarrow s_i$ e s_j são não equivalentes.
   - Se $NEq[i,j]$ recebe 1 no passo 2, então s_i e s_j são não equivalentes.
   - $NEq[i,j]$ recebe 1 no passo 8 somente se $\delta(s_i, a) = s_m$ e $\delta(s_j, a) = s_n$, para algum símbolo a, quando o algoritmo já determinou que s_m e s_n são não equivalentes.
     - Se u é a cadeia que mostra a não equivalência de s_m e s_n, então au mostra a não equivalência de s_i e s_j.

□

Estados equivalentes

Teorema 1.38

- *Os estados s_i e s_j são não equivalentes se, e somente se, $NEq[i,j] = 1$ ao término do Algoritmo 1.*

Demonstração.

2. s_i e s_j não equivalentes $\Rightarrow NEq[i,j] = 1$.
   - Indução no comprimento da menor cadeia que mostra a não equivalência de um par de estados.

□

Estados equivalentes

Teorema 1.38

- *Os estados s_i e s_j são não equivalentes se, e somente se, $NEq[i,j] = 1$ ao término do Algoritmo 1.*

Demonstração.

2. s_i e s_j não equivalentes $\Rightarrow NEq[i,j] = 1$.

Base: Pares de estados que são não equivalentes por uma cadeia de comprimento 0.

- $\overline{\delta}(s_i, \varepsilon) = s_i$ e $\overline{\delta}(s_j, \varepsilon) = s_j$.
- Apenas um entre s_i e s_j é estado final e $NEq[i,j]$ recebe 1 no passo 2.

□

Estados equivalentes

Teorema 1.38

- *Os estados s_i e s_j são não equivalentes se, e somente se, $NEq[i,j] = 1$ ao término do Algoritmo 1.*

Demonstração.

2. s_i e s_j não equivalentes $\Rightarrow NEq[i,j] = 1$.

Hipótese: Suponha que todo par de estados não equivalentes por uma cadeia de comprimento k ou menor é marcado pelo algoritmo.

□

Estados equivalentes

Teorema 1.38

- *Os estados s_i e s_j são não equivalentes se, e somente se, $NEq[i, j] = 1$ ao término do Algoritmo 1.*

Demonstração.

2. s_i e s_j não equivalentes $\Rightarrow NEq[i, j] = 1$.

Passo: Sejam s_i e s_j estados para os quais a menor cadeia u que os tornam não equivalentes é de comprimento $k + 1$.

- $u = av$.
- $\overline{\delta}(s_i, u) = \overline{\delta}(s_i, av) = \overline{\delta}(s_m, v) = s_p$.
- $\overline{\delta}(s_j, u) = \overline{\delta}(s_j, av) = \overline{\delta}(s_n, v) = s_q$.
- Apenas um entre s_p e s_q é estado final, já que o processamento precedente torna s_i e s_j não equivalentes.

$\square$

Estados equivalentes

Teorema 1.38

- *Os estados s_i e s_j são não equivalentes se, e somente se, $NEq[i, j] = 1$ ao término do Algoritmo 1.*

Demonstração.

2. s_i e s_j não equivalentes $\Rightarrow NEq[i, j] = 1$.

Passo:
- Claramente, s_m e s_n são não equivalentes por uma cadeia de comprimento k.
- Por H.I., $NEq[m, n] = 1$.
- Se $NEq[m, n]$ é marcado antes dos estados s_i e s_j serem examinados no passo 5, então $NEq[i, j]$ recebe 1 por uma chamada ao procedimento $DIST$.

$\square$

Estados equivalentes

Teorema 1.38

- *Os estados s_i e s_j são não equivalentes se, e somente se, $NEq[i, j] = 1$ ao término do Algoritmo 1.*

Demonstração.

2. s_i e s_j não equivalentes $\Rightarrow NEq[i, j] = 1$.

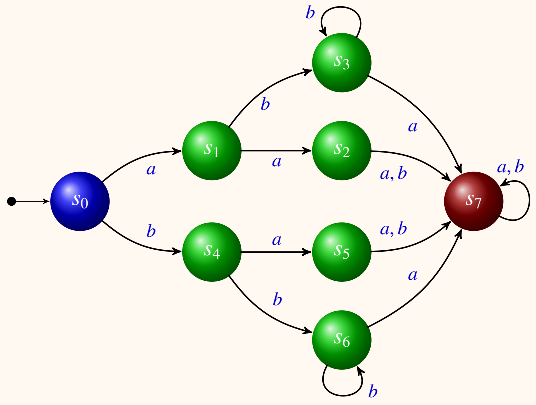
Passo:
- Se s_i e s_j são examinados no passo 8 e $NEq[i, j] \neq 1$, então (i, j) é inserido no conjunto $Dep[m, n]$.
- Por H.I., $NEq[m, n]$ eventualmente recebe 1 por uma chamada a $DIST(m, n)$.
- $NEq[i, j]$ também recebe 1 pela chamada recursiva a $DIST$, já que $(i, j) \in Dep[m, n]$.

$\square$

Exemplos de Minimização de DFA's

Exemplo 1.39

- $\mathcal{L}(M_1) = (a \cup b)(a \cup b^*)$:



Exemplos de Minimização de DFA's

Exemplo 1.39

- Índices marcados com 1 no passo 2:
  - $NEq[0,1]$, $NEq[0,2]$, $NEq[0,3]$, $NEq[0,4]$, $NEq[0,5]$, $NEq[0,6]$, $NEq[1,7]$, $NEq[2,7]$, $NEq[3,7]$, $NEq[4,7]$, $NEq[5,7]$ e $NEq[6,7]$.

Exemplos de Minimização de DFA's

Exemplo 1.39

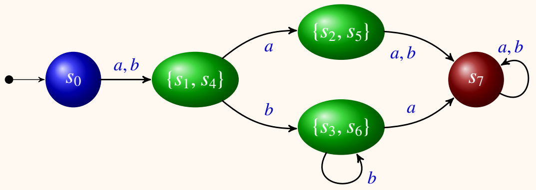
- Cada índice não marcado recebe 1 no passo 8:

Índices	Ação	Motivo
0,7	$NEq[0,7] = 1$	não equivalente por a
1,2	$NEq[1,2] = 1$	não equivalente por a
1,3	$NEq[1,3] = 1$	não equivalente por a
1,4	$Dep[2,5] = \{(1,4)\}$	
	$Dep[3,6] = \{(1,4)\}$	
1,5	$NEq[1,5] = 1$	não equivalente por a
1,6	$NEq[1,6] = 1$	não equivalente por a
2,3	$NEq[2,3] = 1$	não equivalente por b
2,4	$NEq[2,4] = 1$	não equivalente por a
2,5		sem ação, pois $\delta(s_2, x) = \delta(s_5, x)\ \forall\ x \in \Sigma$
2,6	$NEq[2,6] = 1$	não equivalente por b
3,4	$NEq[3,4] = 1$	não equivalente por a
3,5	$NEq[3,5] = 1$	não equivalente por b
3,6		
4,5	$NEq[4,5] = 1$	não equivalente por a
4,6	$NEq[4,6] = 1$	não equivalente por a
5,6	$NEq[5,6] = 1$	não equivalente por b

Exemplos de Minimização de DFA's

Exemplo 1.39

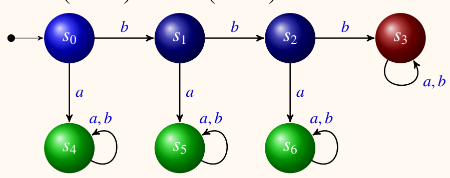
- Após todos os pares de índices serem examinados, apenas os pares de estados $[1,4]$, $[2,5]$ e $[3,6]$ são equivalentes:



Exemplos de Minimização de DFA's

Exemplo 1.40

- $\mathcal{L}(M_1) = a(a \cup b)^* \cup ba(a \cup b)^* \cup bba(a \cup b)^*$



Exemplos de Minimização de DFA's

Exemplo 1.40

- Índices marcados com 1 no passo 2:
  - $NEq[0,4]$, $NEq[0,5]$, $NEq[0,6]$, $NEq[1,4]$, $NEq[1,5]$, $NEq[1,6]$, $NEq[2,4]$, $NEq[2,5]$, $NEq[2,6]$, $NEq[3,4]$, $NEq[3,5]$ e $NEq[3,6]$.

Exemplos de Minimização de DFA's

Exemplo 1.40

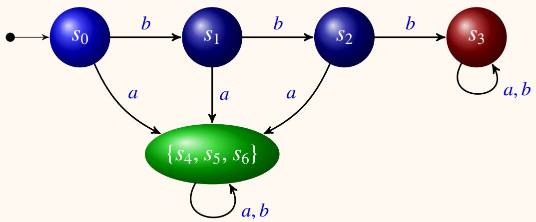
- Cada índice não marcado recebe 1 no passo 8:

Índices	Ação	Motivo
0,1	$Dep[4,5] = \{(0,1)\}$	
	$Dep[1,2] = \{(0,1)\}$	
0,2	$Dep[4,6] = \{(0,2)\}$	
	$Dep[1,3] = \{(0,2)\}$	
0,3	$NEq[0,3] = 1$	não equivalente por a
1,2	$Dep[5,6] = \{(1,2)\}$	
	$Dep[2,3] = \{(1,2)\}$	
1,3	$NEq[1,3] = 1$	não equivalente por a
	$NEq[0,2] = 1$	chamada a $DIST(1,3)$
2,3	$NEq[2,3] = 1$	não equivalente por a
	$NEq[1,2] = 1$	chamada a $DIST(1,2)$
	$NEq[0,1] = 1$	chamada a $DIST(0,1)$
4,5		
4,6		
5,6		

Exemplos de Minimização de DFA's

Exemplo 1.40

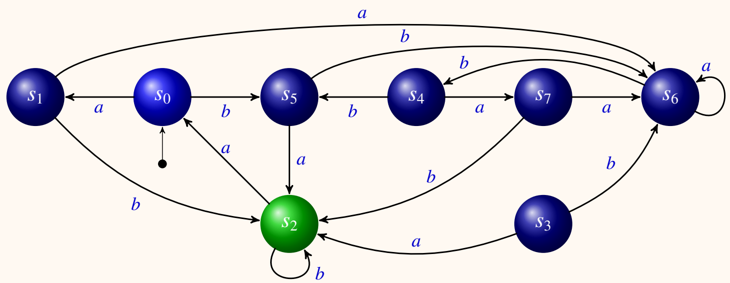
- Após todos os pares de índices serem examinados, as posições $[4,5]$, $[4,6]$ e $[5,6]$ na matriz NEq indicam pares de estados equivalentes. Logo, os pares de estados (s_4, s_5), (s_4, s_6) e (s_5, s_6) são, individualmente, equivalentes. Portanto, s_4, s_5 e s_6 são todos equivalentes entre si:



Exemplos de Minimização de DFA's

Exemplo 1.41

- $\mathcal{L}(M_1) = ???$



Exemplos de Minimização de DFA's

Exemplo 1.41

1. Construir uma matriz triangular superior para pares de estados do autômato.
   - Os pares de estados (s_i, s_j) e (s_j, s_i) são representados na mesma posição da matriz ($NEq[i, j]$ se $i < j$ ou $NEq[j, i]$ se $j < i$).
   - Posições da matriz inicialmente são vazias.

Exemplos de Minimização de DFA's

Exemplo 1.41

2. Marcar com um "1" toda posição $[i, j]$, com $i < j$, onde s_i é um estado final e s_j não é final, ou vice-versa.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7
s_0		1					
s_1		1					
s_2			1	1	1	1	1
s_3							
s_4							
s_5							
s_6							

Exemplos de Minimização de DFA's

Exemplo 1.41

3. Criar uma lista vazia de pares de estados para cada posição não marcada na matriz.
4. Para cada posição $[i, j]$ não marcada, verificar se a posição $[\delta(s_i, a), \delta(s_j, a)]$ está marcada para algum símbolo $a \in \Sigma$.
   - Em caso positivo, marcar a posição $[i, j]$.
   - Em caso negativo, adicionar (i, j) às listas associadas às posições $[\delta(s_i, a), \delta(s_j, a)]$, com $a \in \Sigma$.

   - $[\delta(s_0, a), \delta(s_1, a)] = [s_1, s_6] \Rightarrow$ não marcada; e
     $[\delta(s_0, b), \delta(s_1, b)] = [s_5, s_2] \Rightarrow$ marcada.
     Conclusão: marcar também $[s_0, s_1]$.

Exemplos de Minimização de DFA's

Exemplo 1.41

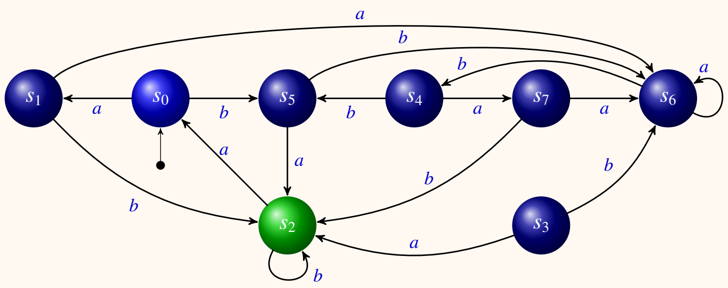
5. Quando a posição $[i, j]$ for marcada, marcar também todas as posições na lista de $[i, j]$.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7
s_0	1	1	1		1	1	1
s_1		1	1	1	1	1	
s_2			1	1	1	1	1
s_3				1		1	1
s_4					1	1	1
s_5						1	1
s_6							1

Exemplos de Minimização de DFA's

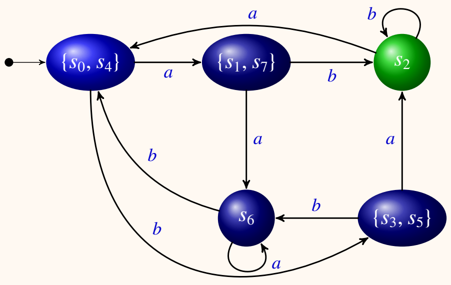
Exemplo 1.41

6. Colapsar os pares de estados que não foram marcados.
7. Eliminar estados não alcançáveis.



Exemplos de Minimização de DFA's

Exemplo 1.41

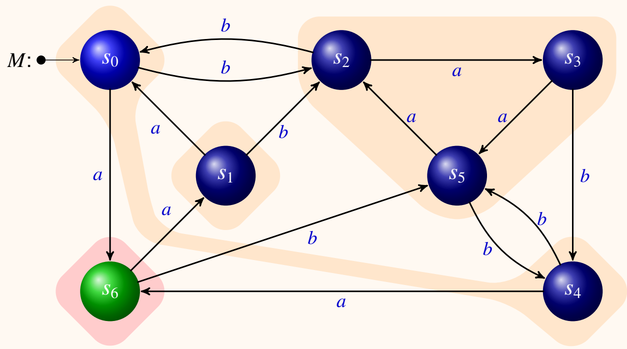


Minimização de DFA's – II

- Dado um DFA $M = \langle \Sigma, S, s_0, \delta, F \rangle$, é possível construir a partir de M um DFA $M' = \langle \Sigma, S', s'_0, \delta', F' \rangle$, com número mínimo de estados, que também reconheça a linguagem $\mathcal{L}(M)$:
  1. Particionar o conjunto S de estados em dois subconjuntos. Um conjunto com todos os estados de aceitação e o outro com estados de não aceitação.
  2. Iterativamente, particionar cada subconjunto da partição corrente: se dois estados de um conjunto forem não equivalentes, dividir esse subconjunto.
  3. Mesclar todos os estados de um conjunto em um novo estado. O número de estados no DFA minimizado é igual ao número de subconjuntos na última partição do conjunto S.

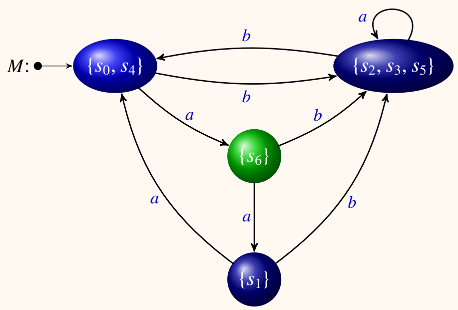
Minimização de DFA's – II

Exemplo 1.42



Minimização de DFA's – II

Exemplo 1.42



Minimização de DFA's – II

Algoritmo 3: EstadosEquivalentes(M)

`// Determina os estados equivalentes de um DFA M`

Entrada: DFA $M = \langle \Sigma, S, s_0, \delta, F \rangle$.
Saída: Partição P de S.

$P_0 \leftarrow \{F, S - F\}$;
$k \leftarrow 1$;
$P_1 \leftarrow \emptyset$;
enquanto $(P_k \neq P_{k-1})$ **faça**
    $T \leftarrow \emptyset$;
    para todo $p \in P_k$ **faça**
        $T \leftarrow T \cup$ Particione(p);
    $k \leftarrow k + 1$;
    $P_k \leftarrow T$;
retorna P_k.

Minimização de DFA's – II

Algoritmo 4: Particione(P)

`// Particiona o conjunto P em subconjuntos de estados equivalentes`

Entrada: Subconjunto p de uma partição S.
Saída: Partionamento de p.

1 **para todo** $(a \in \Sigma)$ **faça**
2   **se** *a particiona p em p_1 e p_2* **então**
3   **retorna** p_1, p_2
4 **retorna** p

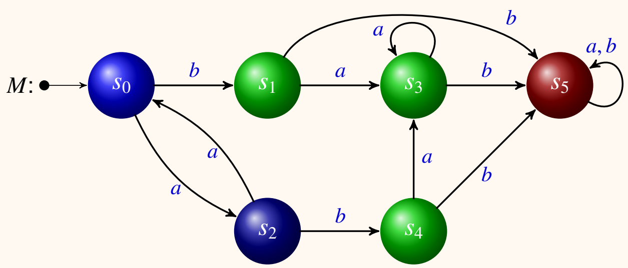
Linha 2 do Algoritmo 4

- Dados dois estados $s_i, s_j \in p$ $(p \in P)$, um símbolo $a \in \Sigma$ particiona p, se:
  1. $\delta(s_i, a) = s_m$,
  2. $\delta(s_j, a) = s_n$,
  3. $s_m \in p_k$, $s_n \in p_\ell$ e $k \neq \ell$
     (s_m e s_n pertencem a subconjuntos diferentes da partição P).

Exemplos de minimização de DFA's

Exemplo 1.43

- $M = \langle \Sigma = \{a, b\}, S = \{s_0, s_1, s_2, s_3, s_4, s_5\}, s_0, \delta, F = \{s_1, s_3, s_4\} \rangle$.
- $\mathcal{L}(M) = ???$



Exemplos de minimização de DFA's

Exemplo 1.43

- $\mathcal{L}(M) = a^*ba^*$!!!
- $P_0 = \{p_1^0 = F, p_2^0 = S - F\} = \{p_1^0 = \{s_1, s_3, s_4\}, p_2^0 = \{s_0, s_2, s_5\}\}$.

Exemplos de minimização de DFA's

Exemplo 1.43

- Para calcular P_1, deve-se verificar se os subconjuntos de P_0 podem ser particionados.
- $p_1^0 = \{s_1, s_3, s_4\}$:
  - $\delta(s_1, a) = \delta(s_3, a) = s_3$ e $\delta(s_1, b) = \delta(s_3, b) = s_5 \Rightarrow s_1$ e s_3 são equivalentes.
  - $\delta(s_1, a) = \delta(s_4, s) = s_3$ e $\delta(s_1, b) = \delta(s_4, b) = s_5 \Rightarrow s_1$ e s_4 são equivalentes.
  - Como os pares (s_1, s_3) e (s_1, s_4) são equivalentes, então (s_3, s_4) também são equivalentes. Logo, p_1^0 não será particionado em P_1, ou seja, $p_1^1 = p_1^0$.
- $p_2^0 = \{s_0, s_2, s_5\}$:
  - $\delta(s_0, a) = s_2$ e $\delta(s_2, a) = s_0$ $(s_0, s_2 \in p_2^0)$; $\delta(s_0, b) = s_1$ e $\delta(s_2, b) = s_4$ $(s_1, s_4 \in p_1^0) \Rightarrow s_0$ e s_2 são equivalentes.
  - $\delta(s_0, a) = s_2$ e $\delta(s_5, a) = s_5$ $(s_2, s_5 \in p_2^0)$; $\delta(s_0, b) = s_1$ e $\delta(s_5, b) = s_5$ $(s_1 \in p_1^0$ e $s_5 \in p_2^0) \Rightarrow s_0$ e s_5 não são equivalentes.
  - Como (s_0, s_2) são equivalentes e (s_0, s_5) não são equivalentes, então (s_2, s_5) também não são equivalentes. Logo, p_2^0 será particionado em $p_2^1 = \{s_0, s_2\}$ e $p_3^1 = \{s_5\}$.
- $P_1 = \{p_1^1 = \{s_1, s_3, s_4\}, p_2^1 = \{s_0, s_2\}, p_3^1 = \{s_5\}\}$.

Exemplos de minimização de DFA's

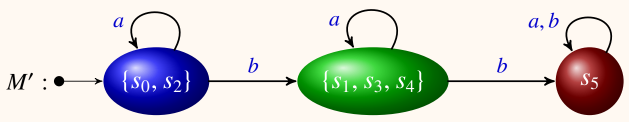
Exemplo 1.43

- Para calcular P_2, deve-se verificar se os subconjuntos de P_1 podem ser particionados.
- $p_1^1 = \{s_1, s_3, s_4\}$:
  - Como os pares (s_1, s_3) e (s_1, s_4) são equivalentes, então (s_3, s_4) também são equivalentes (argumentação idêntica à usada para o subconjunto p_1^0). Logo, p_1^1 não será particionado em P_1, ou seja, $p_1^2 = p_1^1$.
- $p_2^1 = \{s_0, s_2\}$:
  - $\delta(s_0, a) = s_2$ e $\delta(s_2, a) = s_0$ $(s_0, s_2 \in p_2^1)$; $\delta(s_0, b) = s_1$ e $\delta(s_2, b) = s_4$ $(s_1, s_4 \in p_1^1) \Rightarrow s_0$ e s_2 são equivalentes.
  - Logo, p_2^1 não será particionado em P_2, ou seja, $p_2^2 = p_2^1$.
- $p_3^1 = \{s_5\}$:
  - Como há apenas um estado nesse conjunto, ele não pode ser mais particionado, ou seja, $p_3^2 = p_3^1$.
- $P_2 = \{p_1^2 = \{s_1, s_3, s_4\}, p_2^2 = \{s_0, s_2\}, p_3^2 = \{s_5\}\} = P_1$.

Exemplos de minimização de DFA's

Exemplo 1.43

- Dado que $P_1 = P_2$, esta é a partição final.
- A partição P_2 significa que os estados s_1, s_3 e s_4 serão mesclados em apenas um estado. Da mesma forma, s_0 e s_2 serão mesclados em um novo estado.
- DFA M' minimizado e correspondente ao DFA M:



Livros texto

R. P. Grimaldi
Discrete and Combinatorial Mathematics – An Applied Introduction.
Addison Wesley, 1994.

D. J. Velleman
How To Prove It – A Structured Approach.
Cambridge University Press, 1996.

J. E. Hopcroft; J. Ullman.
Introdução Ã Teoria de Autômatos, Linguagens e Computação.
Ed. Campus.

T. A. Sudkamp.
Languages and Machines – An Introduction to the Theory of Computer Science.
Addison Wesley Longman, Inc. 1998.

J. Carroll; D. Long.
Theory of Finite Automata – With an Introduction to Formal Languages.
Prentice-Hall, 1989.

M. Sipser.
Introduction to the Theory of Computation.
PWS Publishing Company, 1997.

H. R. Lewis; C. H. Papadimitriou
Elementos de Teoria da Computação.
Bookman, 2000.