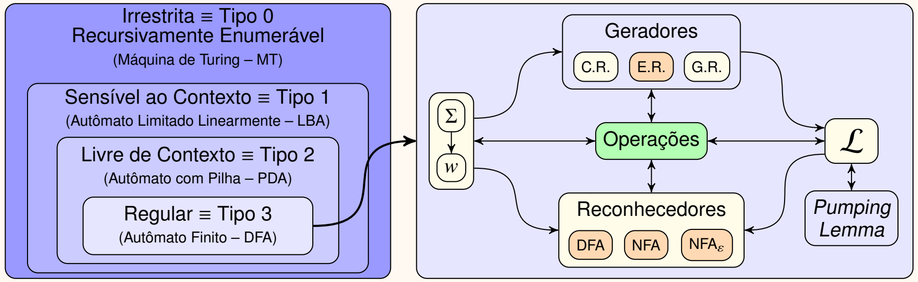
Linguagens Formais e Autômatos

Humberto Longo

Instituto de Informática
Universidade Federal de Goiás

Bacharelado em Ciência da Computação, 2021/1

Roteiro



Equivalência com autômato finito

Lema 5.34

- *Se uma linguagem $\mathcal{L}$, sobre um alfabeto Σ, é regular, então alguma expressão regular $\mathcal{R}$ a descreve.*

Demonstração.

- $\mathcal{L}$ regular $\Rightarrow$ $\mathcal{L}$ é reconhecida por algum NFA.
- Conversão de NFA numa expressão regular equivalente:
  1. conversão de NFA em $GNFA$.
  2. conversão de $GNFA$ em expressão regular.

  $GNFA$: *Generalized Nondeterministic Finite Automaton.*

□

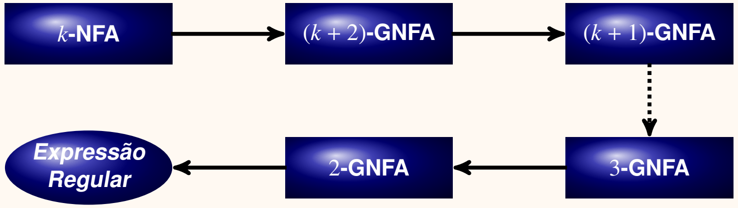
Equivalência com autômato finito

Lema 5.34

- *Se uma linguagem $\mathcal{L}$, sobre um alfabeto Σ, é regular, então alguma expressão regular $\mathcal{R}$ a descreve.*

Demonstração.

- Conversão de NFA numa expressão regular equivalente.
  - k-NFA : autômato finito determinístico com k estados.
  - ℓ-GNFA : autômato finito não determinístico generalizado com ℓ estados.



□

NFA generalizado – $GNFA$

Definição 5.35 (Geral)

- Um $GNFA$ é um NFA onde:
  - função de transição δ aceita como parâmetros um estado e uma expressão regular e tem como resultado outro estado;
  - pode haver transições do estado inicial para todos os demais;
  - sem transições dos demais estados para o estado inicial;
  - há apenas um estado final;
  - pode haver transições dos demais estados para o estado final;
  - sem transição do estado inicial (final) para o próprio; e
  - pode haver transições de cada estado para o próprio (exceto o inicial e o final).

NFA generalizado – $GNFA$

Definição 5.36

- NFA definido pela quíntupla $\langle \Sigma, S, s_0, \delta, F \rangle$.
- Um $GNFA$ é definido pela quíntupla $\langle \Sigma, S, s_{ini}, \delta', s_{fim} \rangle$, onde:

Σ : o alfabeto de entrada;
$S \neq \emptyset$: o conjunto finito de estados;
$s_{ini} \in S$: estado inicial;
$\delta' : (S - \{s_{fim}\}) \times \mathcal{R}^* \to S - \{s_{ini}\}$: a função de transição que associa um estado e uma expressão regular a outro estado;
$s_{fim} \in S$: estado final;

$\mathcal{R}^*$: conjunto de todas as expressões regulares sobre o alfabeto Σ.

NFA generalizado – $GNFA$

Definição 5.37

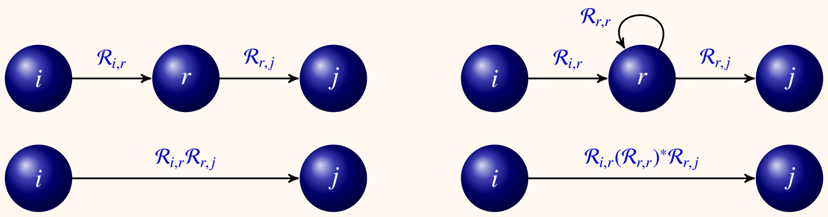
- Alfabeto Σ.
- Cadeia $w = w_1 w_2 \dots w_k$, onde cada $w_i \in \Sigma^*$.
- $GNFA\ G = \langle \Sigma, S, s_{ini}, \delta, s_{fim} \rangle$.
- G reconhece a cadeia w se existe uma sequência de estados $s_1, s_2, \dots, s_k$ tal que:
  - $s_1 = s_{ini}$ é o estado inicial,
  - $s_k = s_{fim}$ é o estado final,
  - $\mathcal{R}_{i-1,i} = w_i$, para cada i, $2 \leqslant i \leqslant k$, onde $\delta(s_{i-1}, \mathcal{R}_{i-1,i}) = s_i$.

Conversão de NFA em $GNFA$

- $NFA = \langle \Sigma, S, s_0, \delta_1, F \rangle$.
- $GNFA = \langle \Sigma, S \cup \{s_{ini}, s_{fim}\}, s_{ini}, \delta, s_{fim} \rangle$.
  - Novo estado s_{ini}, com transição $\delta(s_{ini}, \varepsilon) = s_0$.
  - Novo estado s_{fim}, com transições $\delta(s, \varepsilon) = s_{fim}$, $\forall\ s \in F$.
  - União de múltiplas transições entre estados:
    - $\delta(s_i, a) = s_j$ e $\delta(s_i, b) = s_j \Rightarrow \delta(s_i, a \cup b) = s_j$.
    - Como considerar estado intermediário entre s_i e s_j?

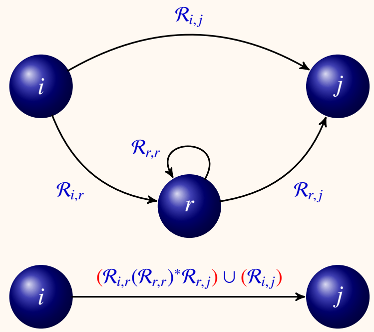
Expressão regular e autômato finito

- Resumo das operações de exclusão de um vértice:



Expressão regular e autômato finito

- Resumo das operações de exclusão de um vértice:



Conversão de $GNFA$ em expressão regular

Algoritmo 7: ExtraiExprReg(G)

Entrada: $GNFA\ G = \langle \Sigma, S, s_{ini}, \delta, s_{fim} \rangle$.
Saída: Expressão regular $\mathcal{R}$

$k \leftarrow |S|$;
se $(k = 2)$ **então**
  $\delta(s_{ini}, \mathcal{R}) = s_{fim}$;
  retorna $\mathcal{R}$;
senão
  $s_r \leftarrow s \in S$, tal que $s \neq s_{ini}$ e $s \neq s_{fim}$.;
  $S' \leftarrow S - \{s_r\}$;
  $\delta'(s_i, \mathcal{R}_{i,r}(\mathcal{R}_{r,r})^*\mathcal{R}_{r,j} \cup \mathcal{R}_{i,j}) \leftarrow s_j, \ \forall \ s_i \in S' - \{s_{ini}\}$ e $s_j \in S' - \{s_{fim}\}$, onde: $\begin{cases} \delta(s_i, \mathcal{R}_{i,r}) = s_r \\ \delta(s_r, \mathcal{R}_{r,r}) = s_r \\ \delta(s_r, \mathcal{R}_{r,j}) = s_j \\ \delta(s_i, \mathcal{R}_{i,j}) = s_j. \end{cases}$
ExtraiExprReg($G' = \langle \Sigma, S', s_{ini}, \delta', s_{fim} \rangle$);

Conversão de $GNFA$ em expressão regular

Lema 5.38

- *Seja G um $GNFA$ e G' o $GNFA$ obtido a partir de G em uma iteração do Algoritmo 7, então G e G' são equivalentes.*

Demonstração.

- Indução no número k de estados de G':

Base: $k = 2 \Rightarrow S = \{s_{ini}, s_{fim}\} \Rightarrow \delta(s_{ini}, \mathcal{R}) = s_{fim}$.

□

Conversão de $GNFA$ em expressão regular

Lema 5.38

- *Seja G um $GNFA$ e G' o $GNFA$ obtido a partir de G em uma iteração do Algoritmo 7, então G e G' são equivalentes.*

Demonstração.

- Indução no número k de estados de G':

Passo 1: Suponha que G aceita uma cadeia w:

- $\exists\ s_{ini} \equiv s_1, s_2, s_3, \ldots, s_q \equiv s_{fim}$
- $s_r \neq s_2, s_3, \ldots, s_{q-1} \Rightarrow G'$ aceita w
- $s_r = s_p, 2 \leqslant p \leqslant q-1 \Rightarrow \delta'(s_i, \mathcal{R}') = s_j$ descreve w (G' aceita w).

$\square$

Conversão de $GNFA$ em expressão regular

Lema 5.38

- *Seja G um $GNFA$ e G' o $GNFA$ obtido a partir de G em uma iteração do Algoritmo 7, então G e G' são equivalentes.*

Demonstração.

- Indução no número k de estados de G':

Passo 2: Suponha que G' aceita uma cadeia w:

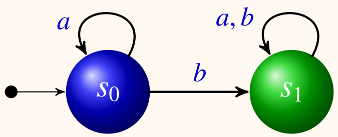
- $\delta'(s_i, \mathcal{R}') = s_j) \Rightarrow \mathcal{R}'$ descreve coleção de cadeias reconhecidas entre s_i e s_j em G, passando ou não por s_r.

G e G' são equivalentes.

$\square$

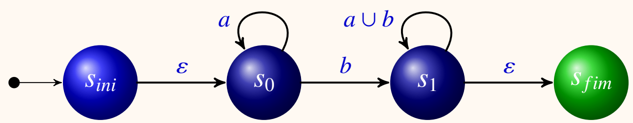
Conversão de $GNFA$ em expressão regular

Exemplo 5.39



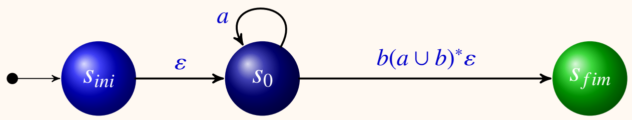
Conversão de $GNFA$ em expressão regular

Exemplo 5.39



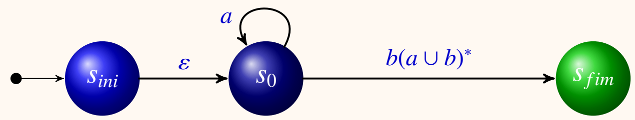
Conversão de $GNFA$ em expressão regular

Exemplo 5.39



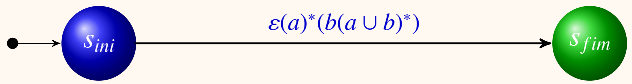
Conversão de $GNFA$ em expressão regular

Exemplo 5.39



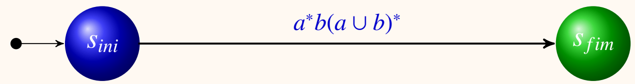
Conversão de $GNFA$ em expressão regular

Exemplo 5.39



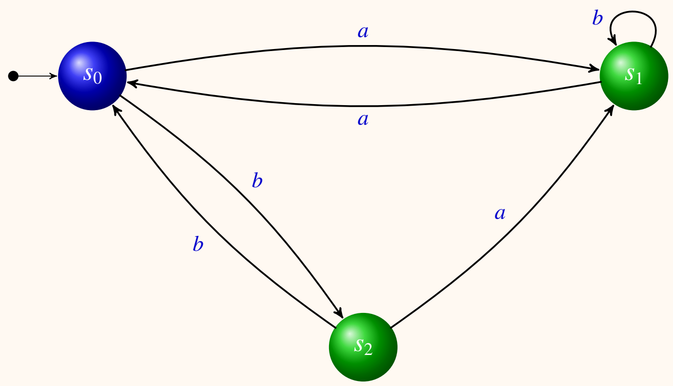
Conversão de $GNFA$ em expressão regular

Exemplo 5.39



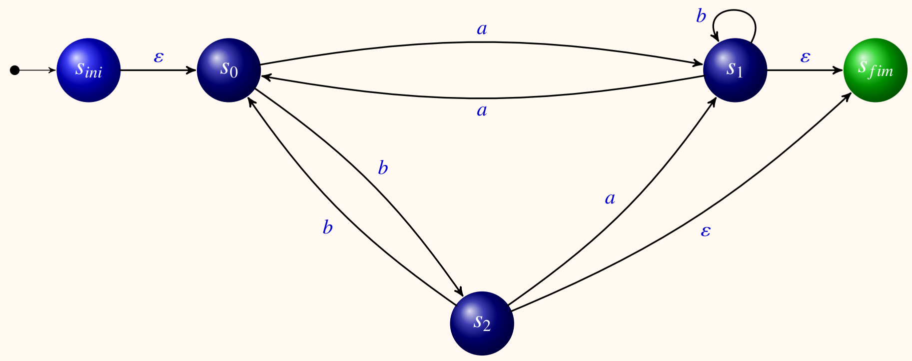
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



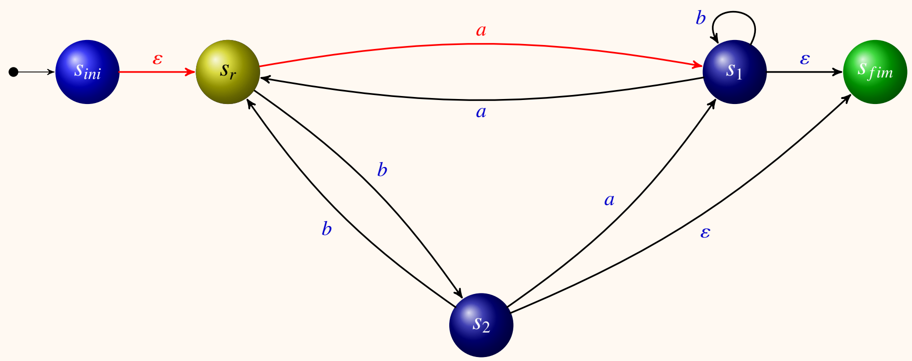
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



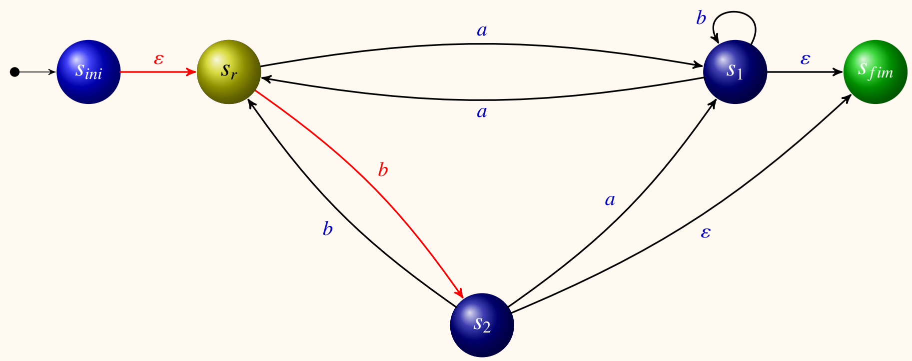
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



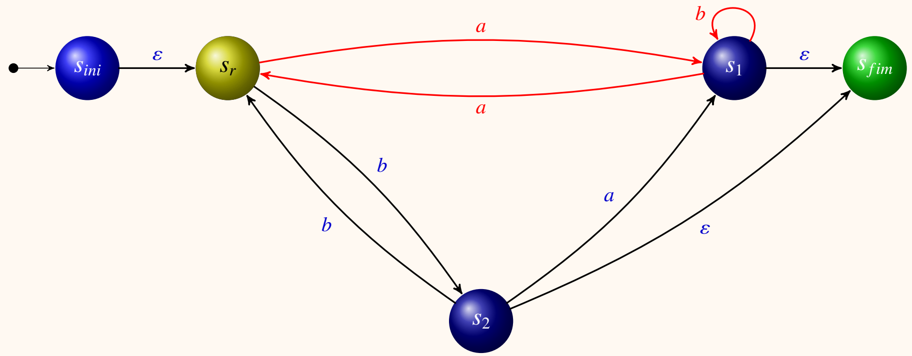
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



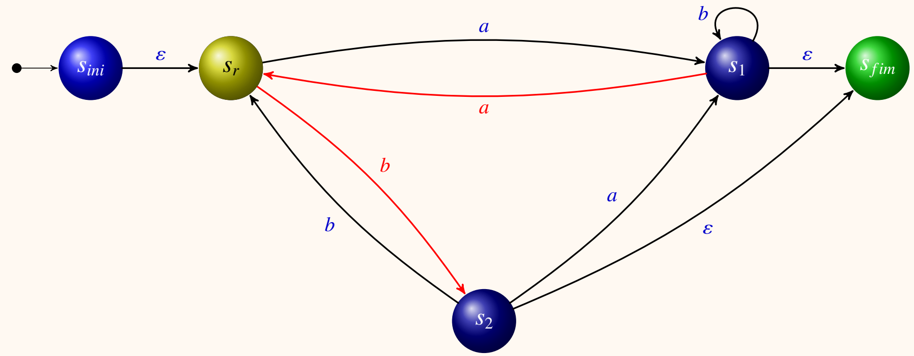
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



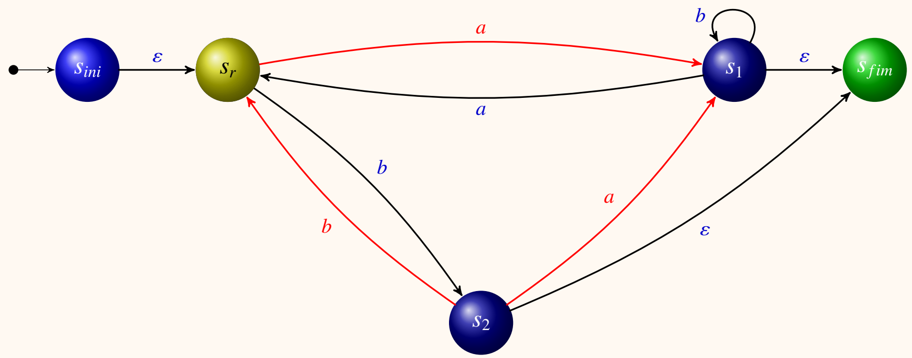
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



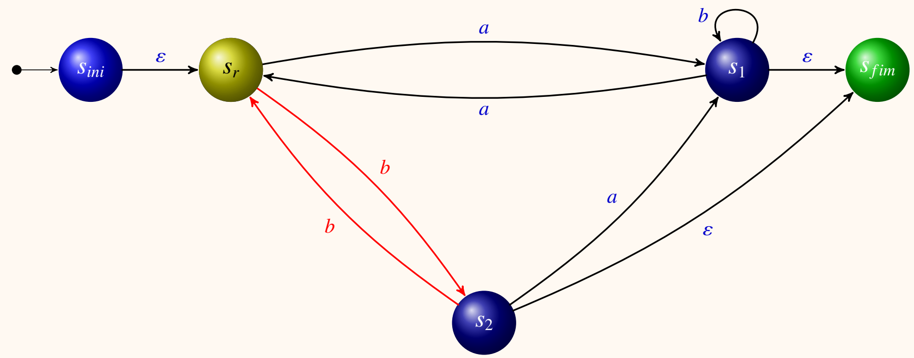
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



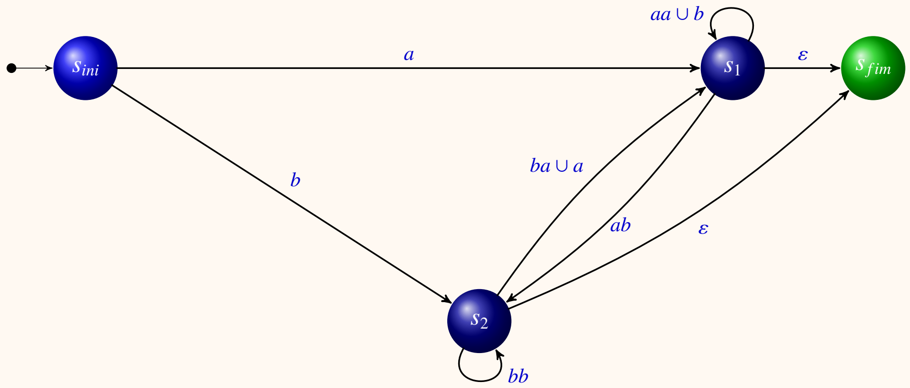
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



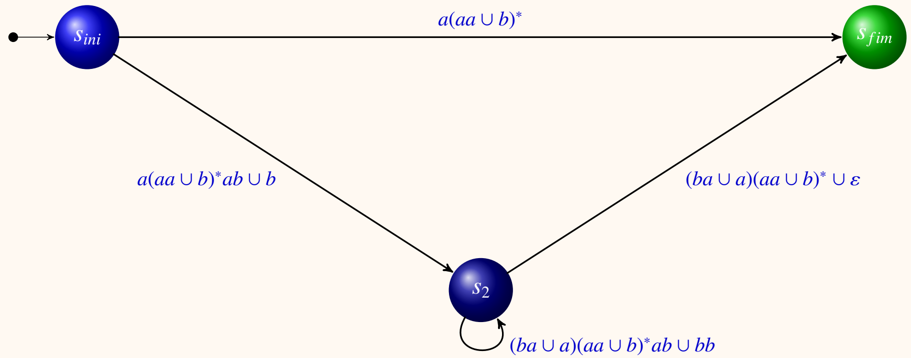
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



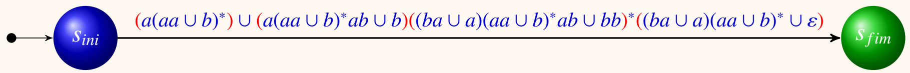
Conversão de $GNFA$ em expressão regular

Exemplo 5.40



Conversão de $GNFA$ em expressão regular

Exemplo 5.40



Conversão de NFA em expressão regular

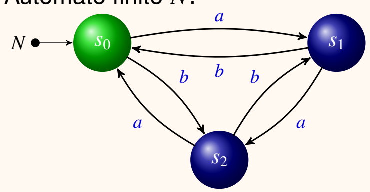
- A mesma técnica descrita para converter gramática regular em expressão regular pode ser usada a partir de um NFA!
- Seja o NFA $N = \langle \Sigma, S = \{s_0, s_1, \ldots, s_n\}, s_0, \delta, F \rangle$ sem transições ε.
- Definem-se expressões $\mathcal{S}_0, \ldots, \mathcal{S}_n$, de modo que cada $\mathcal{S}_i$ gere todas as cadeias que são aceitas por N a partir do estado s_i. Obviamente, $\mathcal{L}(\mathcal{S}_0) = \mathcal{L}(N)$.
- Sistema de equações:

$$\mathcal{S}_i = \bigcup_{\delta(s_i,a)=s_j} a\mathcal{S}_j \quad \bigcup_{s_i \in F} \varepsilon.$$

Autômato finito e sistema de equações

Exemplo 5.41

- Autômato finito N:



- Sistema de equações:

$$\mathcal{S}_0 = a\mathcal{S}_1 \cup b\mathcal{S}_2 \cup \varepsilon \qquad (1)$$
$$\mathcal{S}_1 = a\mathcal{S}_2 \cup b\mathcal{S}_0 \qquad (2)$$
$$\mathcal{S}_2 = a\mathcal{S}_0 \cup b\mathcal{S}_1 \qquad (3)$$

Autômato finito e sistema de equações

Exemplo 5.41

Solução :

$(3) \rightarrow (1)$ e $(3) \rightarrow (2)$:

$$\begin{aligned} \mathcal{S}_0 &= a\mathcal{S}_1 \cup b(a\mathcal{S}_0 \cup b\mathcal{S}_1) \cup \varepsilon \\ &= (a \cup bb)\mathcal{S}_1 \cup ba\mathcal{S}_0 \cup \varepsilon && (4) \\ \mathcal{S}_1 &= a(a\mathcal{S}_0 \cup b\mathcal{S}_1) \cup b\mathcal{S}_0 \\ &= ab\mathcal{S}_1 \cup (aa \cup b)\mathcal{S}_0 && (5) \end{aligned}$$

Autômato finito e sistema de equações

Exemplo 5.41

Solução :

Lema de Arden em (5):

$$\mathcal{S}_0 = (a \cup bb)\mathcal{S}_1 \cup ba\mathcal{S}_0 \cup \varepsilon \qquad (4)$$
$$\mathcal{S}_1 = (ab)^*(aa \cup b)\mathcal{S}_0 \qquad (6)$$

$(6) \rightarrow (4)$:

$$\begin{aligned} \mathcal{S}_0 &= (a \cup bb)(ab)^*(aa \cup b)\mathcal{S}_0 \cup ba\mathcal{S}_0 \cup \varepsilon \\ &= ((a \cup bb)(ab)^*(aa \cup b) \cup ba)\mathcal{S}_0 \cup \varepsilon && (7) \end{aligned}$$

Lema de Arden em (7):

$$\mathcal{S}_0 = ((a \cup bb)(ab)^*(aa \cup b) \cup ba)^* \qquad (8)$$

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.