Teoria da Computação

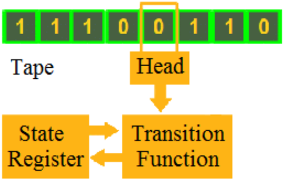
Autômatos Finitos e Linguagens Regulares

Mirtha Lina Fernández Venero
mirtha.lina@ufabc.edu.br

setembro 2017

Sumário

Autômatos Finitos Determinísticos



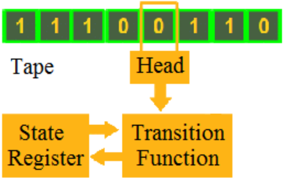
$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F})$

- Q conjunto *finito não vazio* de estados
- Σ alfabeto de entrada
- $q_0 \in Q$ estado inicial
- $\delta : Q \times \Sigma \to Q$ função de transição de estados
- $F \subseteq Q$ conjunto de estados finais

Processo de Reconhecimento:

- A memória do autômato é representada por seus estados. Como Q é finito, **A** só pode lembrar um número finito e fixo de situações durante o reconhecimento da cadeia
- **Determinístico**: em qualquer ponto do reconhecimento, o autômato tem uma única maneira de proceder.

Autômatos Finitos Determinísticos



$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F})$

- Q conjunto *finito não vazio* de estados
- Σ alfabeto de entrada
- $q_0 \in Q$ estado inicial
- $\delta : Q \times \Sigma \to Q$ função de transição de estados
- $F \subseteq Q$ conjunto de estados finais

Processo de Reconhecimento:

- Começar pelo **estado inicial**
- Ler cada símbolo da cadeia e mudar de estado dependendo de δ até chegar no final da cadeia ou não ter como avançar. A cadeia é reconhecida se o último estado é final

Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$$

$$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$$

$$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$$

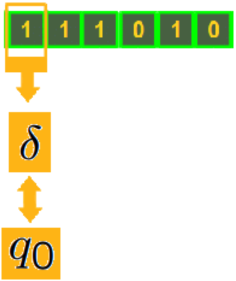
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$



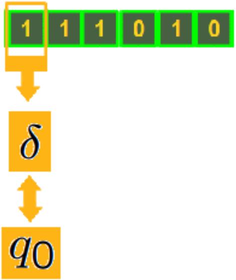
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1$ $\quad$ $\color{red}{\delta(q_0, 1) = q_0}$

$\delta(q_1, 0) = q_1$ $\quad$ $\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2$ $\quad$ $\delta(q_2, 1) = q_2$



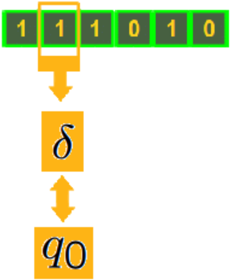
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1$ $\quad$ $\delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1$ $\quad$ $\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2$ $\quad$ $\delta(q_2, 1) = q_2$



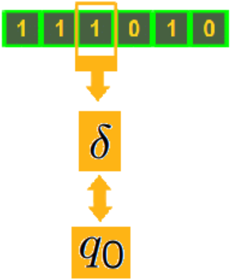
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$



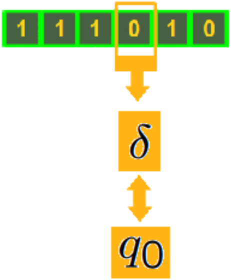
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$



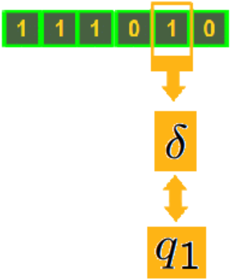
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$



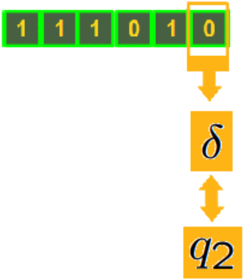
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$

$\color{red}{\delta(q_2, 0) = q_2} \quad \delta(q_2, 1) = q_2$



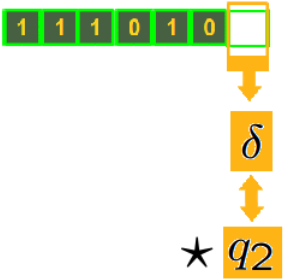
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$$

$$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$$

$$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$$



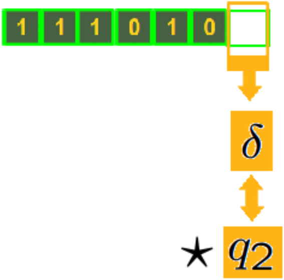
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$$

$$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$$

$$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$$



Extensão da função de transição:

$$\hat{\delta}(q, \epsilon) = q \qquad \hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$$

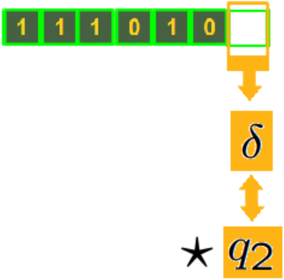
Autômatos Finitos Determinísticos (AFDs)

$$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F}) = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1 \quad \delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$



Extensão da função de transição:

$$\hat{\delta}(q, \epsilon) = q \qquad \hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$$

Linguagem de um AFD: $L(A) = \{w \mid \hat{\delta}(q_0, w) \in F\}$

Autômatos Finitos Determinísticos

Exemplo: Qual a linguagem gerada por A?

$$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0 \quad \delta(q_1, 0) = q_1$$

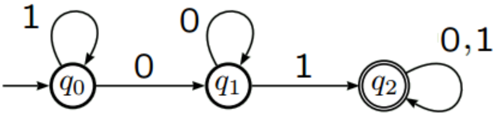
$$\delta(q_1, 1) = q_2 \quad \delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$$

Autômatos Finitos Determinísticos

Exemplo: Qual a linguagem gerada por A?

$$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0 \quad \delta(q_1, 0) = q_1$$

$$\delta(q_1, 1) = q_2 \quad \delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_2$$



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$\star\ q_2$	q_2	q_2

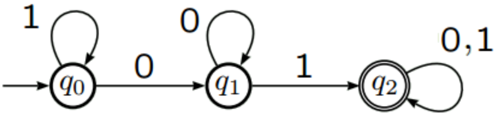
$$q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \ \star$$

Autômatos Finitos Determinísticos

Exemplo: Qual a linguagem gerada por A? $\{\mathbf{0},\mathbf{1}\}^* \cdot \mathbf{01} \cdot \{\mathbf{0},\mathbf{1}\}^*$

$$\delta(q_0,0)=q_1 \quad \delta(q_0,1)=q_0 \quad \delta(q_1,0)=q_1$$

$$\delta(q_1,1)=q_2 \quad \delta(q_2,0)=q_2 \quad \delta(q_2,1)=q_2$$

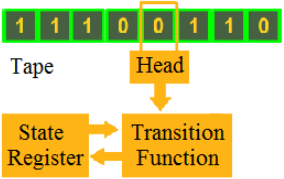


	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$\star\, q_2$	q_2	q_2

$$q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \ \star$$

Sumário

Autômatos Finitos Não Determinísticos



$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F})$

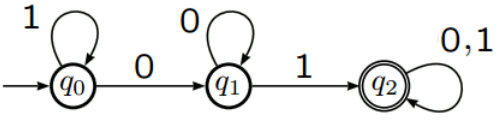
- Q conjunto *finito não vazio* de estados
- Σ alfabeto de entrada
- $q_0 \in Q$ estado inicial
- $\delta : Q \times \Sigma \to \wp(Q)$ função de transição de estados
- $F \subseteq Q$ conjunto de estados finais

Processo de Reconhecimento:

- **Não Determinístico**: o autômato tem várias maneiras de proceder para ao menos um estado e um símbolo. **Como???** **Oracle** (sabe qual caminho tomar), **multi-threading** (prova todos os caminhos simultaneamente), **backtracking** (escolhe um e volta atrás se não der certo)

Autômatos Finitos Não Determinísticos

Exemplo: O conjunto de todas as palavras sobre $\Sigma = \{0, 1\}$ que contém a sub-palavra 01, i.e. $\{\mathbf{0}, \mathbf{1}\}^* \cdot \mathbf{01} \cdot \{\mathbf{0}, \mathbf{1}\}^*$

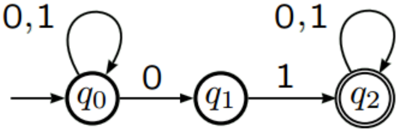


	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$\star\ q_2$	q_2	q_2

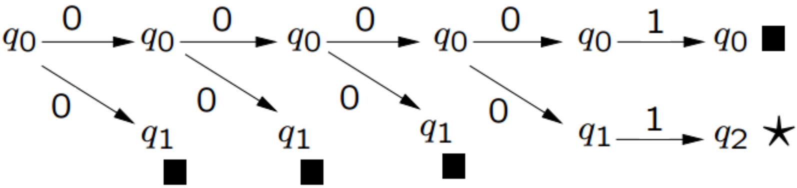
$$q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \star$$

Autômatos Finitos Não Determinísticos

Exemplo: O conjunto de todas as palavras sobre $\Sigma = \{0, 1\}$ que contém a sub-palavra 01, i.e. $\{\mathbf{0}, \mathbf{1}\}^* \cdot \mathbf{01} \cdot \{\mathbf{0}, \mathbf{1}\}^*$

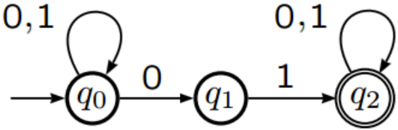


	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$\star\ q_2$	$\{q_2\}$	$\{q_2\}$



Autômatos Finitos Não Determinísticos

Exemplo: O conjunto de todas as palavras sobre $\Sigma = \{0, 1\}$ que contém a sub-palavra 01, i.e. $\{\mathbf{0}, \mathbf{1}\}^* \cdot \mathbf{01} \cdot \{\mathbf{0}, \mathbf{1}\}^*$

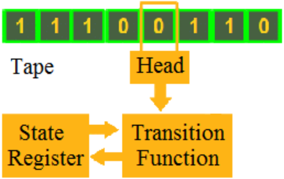


	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$\star\ q_2$	$\{q_2\}$	$\{q_2\}$

$$\{q_0\} \xrightarrow{0} \{q_0, q_1\} \xrightarrow{0} \{q_0, q_1\} \xrightarrow{0} \{q_0, q_1\}$$

$$\star\ \{q_0, q_2\} \xleftarrow{1} \{q_0, q_1\} \xleftarrow{0}$$

Autômatos Finitos Não Determinísticos



$\mathbf{A} = (\mathbf{Q}, \mathbf{\Sigma}, \mathbf{q_0}, \delta, \mathbf{F})$

- Q conjunto *finito não vazio* de estados
- Σ alfabeto de entrada
- $q_0 \in Q$ estado inicial
- $\delta : Q \times \Sigma \to \wp(Q)$ função de transição de estados
- $F \subseteq Q$ conjunto de estados finais

Extensão da função de transição:

$$\hat{\delta}(q, \epsilon) = \{q\} \qquad \hat{\delta}(q, xa) = \bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a)$$

Linguagem de um AFND: $L(A) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$

Universidade Federal do ABC

Autômatos Não Determinísticos com ε-transições

O autômato pode mudar de estado sem considerar (nem consumir) o símbolo de entrada

- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$

Autômatos Não Determinísticos com ε-transições

O autômato pode mudar de estado sem considerar (nem consumir) o símbolo de entrada

- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$

ε-**closure**:

$$q \in \mathsf{ECLOSE}(q)$$

$$r \in \mathsf{ECLOSE}(q) \Leftarrow \ p \in \mathsf{ECLOSE}(q) \text{ and } r \in \delta(p, \epsilon)$$

Autômatos Não Determinísticos com ε-transições

O autômato pode mudar de estado sem considerar (nem consumir) o símbolo de entrada

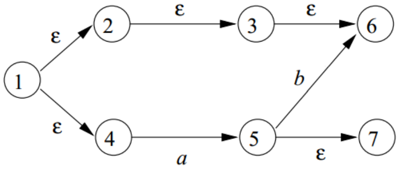
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$

ε-**closure**:

$$q \in \text{ECLOSE}(q)$$

$$r \in \text{ECLOSE}(q) \Leftarrow p \in \text{ECLOSE}(q) \text{ and } r \in \delta(p, \epsilon)$$

Exemplo: ECLOSE(1) =



Autômatos Não Determinísticos com ε-transições

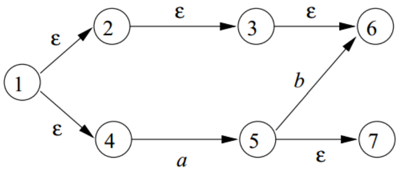
O autômato pode mudar de estado sem considerar (nem consumir) o símbolo de entrada

- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$

ε-**closure**:

$$q \in \mathrm{ECLOSE}(q)$$
$$r \in \mathrm{ECLOSE}(q) \Leftarrow \; p \in \mathrm{ECLOSE}(q) \text{ and } r \in \delta(p, \epsilon)$$

Exemplo: $\mathbf{ECLOSE(1) = \{1, 2, 3, 4, 6\}}$



Autômatos Não Determinísticos com ε-transições

O autômato pode mudar de estado sem considerar (nem consumir) o símbolo de entrada

- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$

ε-**closure**:

$$q \in \text{ECLOSE}(q)$$
$$r \in \text{ECLOSE}(q) \Leftarrow \; p \in \text{ECLOSE}(q) \text{ and } r \in \delta(p, \epsilon)$$

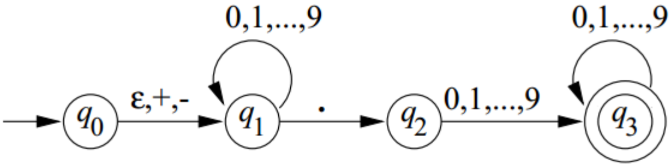
Extensão da função de transição:

$$\hat{\delta}(q, \epsilon) = \text{ECLOSE}(q)$$

$$\hat{\delta}(q, xa) = \bigcup_{p \in \delta(\hat{\delta}(q,x),a)} \text{ECLOSE}(p)$$

Autômatos Não Determinísticos com ε-transições

O autômato pode mudar de estado sem considerar (nem consumir) o símbolo de entrada



Toda linguagem que pode ser reconhecida usando um autômato finito também pode ser definida usando operações de conjuntos, concatenação e fechamento de Kleene (linguagem regular)

Sumário

Expressões regulares

As linguagens regulares são fechadas w.r.t concatenação, $\cap$, $\cup$, ^{c}, $*$ de Kleene. Por isso, uma forma simples de descrever essas linguagens é usar uma notação algébrica (**expressões regulares**)

Linguagem regular	**Expressão regular**
$\emptyset, \{\varepsilon\}, \{a\}, a \in \Sigma$	$\emptyset, \varepsilon, a$
$L_1 \cup L_2$	$ER(L_1) \mid ER(L_2)$
$L_1 L_2$	$ER(L_1)\ ER(L_2)$
L^*	$ER(L)^*$

Expressões regulares

As linguagens regulares são fechadas w.r.t concatenação, $\cap$, $\cup$, ^{c}, $*$ de Kleene. Por isso, uma forma simples de descrever essas linguagens é usar uma notação algébrica (**expressões regulares**)

Exemplo:

Linguagem regular	Expressão regular
$\{a, b, \varepsilon\}$	$a \mid b \mid \varepsilon$
$\{a^n b^m \mid n > 0, m > 0\}$	$aa^* bb^*$
?	$(1 \mid \varepsilon)(01)^*(0 \mid \varepsilon)$
?	$(a \mid b \mid c \mid \ldots \mid z)^*.(txt \mid exe)$

Expressões regulares

As linguagens regulares são fechadas w.r.t concatenação, $\cap$, $\cup$, ^{c}, $*$ de Kleene. Por isso, uma forma simples de descrever essas linguagens é usar uma notação algébrica (**expressões regulares**)

Exemplo:

Linguagem regular	Expressão regular
$\{a, b, \varepsilon\}$	$a \mid b \mid \varepsilon$
$\{a^n b^m \mid n > 0, m > 0\}$	$aa^* bb^*$
?	$(1 \mid \varepsilon)(01)^*(0 \mid \varepsilon)$
?	$(a \mid b \mid c \mid \ldots \mid z)^*.(txt \mid exe)$

Outros padrões podem ser usados: $E?$ para $E \mid \varepsilon$, $E+$ para uma ou mais repetições de E, $E\{n\}$ para n repetições de E, $[a_1 - a_n]$ qualquer símbolo $\in \{a_1, \ldots, a_n\}$

Ver e.g. `http://www.gnu.org/software/grep/manual/grep.html#Fundamental-Structure`

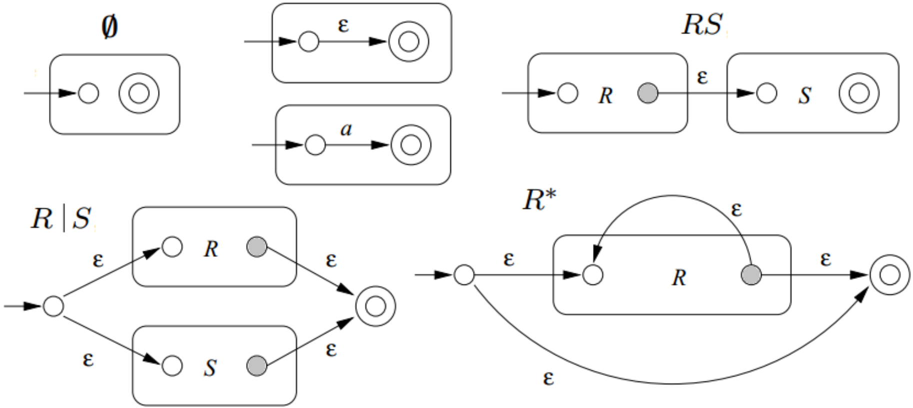
Leis algébricas das expressões regulares

1. $\alpha \mid \beta = \beta \mid \alpha$
2. $\alpha \mid (\beta \mid \gamma) = (\alpha \mid \beta) \mid \gamma$, $\quad \alpha\ (\beta\ \gamma) = (\alpha\ \beta)\ \gamma$
3. $\alpha\ (\beta \mid \gamma) = \alpha\ \beta \mid \alpha\ \gamma$, $\quad (\alpha \mid \beta)\gamma = \alpha\gamma \mid \beta\gamma$
4. $\alpha \mid \emptyset = \emptyset \mid \alpha = \alpha$
5. $\alpha\ \varepsilon = \varepsilon\ \alpha = \alpha$
6. $\alpha\ \emptyset = \emptyset\ \alpha = \emptyset$
7. $\alpha \mid \alpha = \alpha$
8. $\emptyset^* = \emptyset$
9. $(\alpha^*)^* = \alpha^*$
10. $(\alpha^*\ \beta^*)^* = (\alpha \mid \beta)^*$

Exemplo: Simplifique a seguinte ER: $\varepsilon \mid 0 \mid 1 \mid \emptyset\ (\varepsilon \mid 1)^*0$

Linguagens Regulares → Autômatos

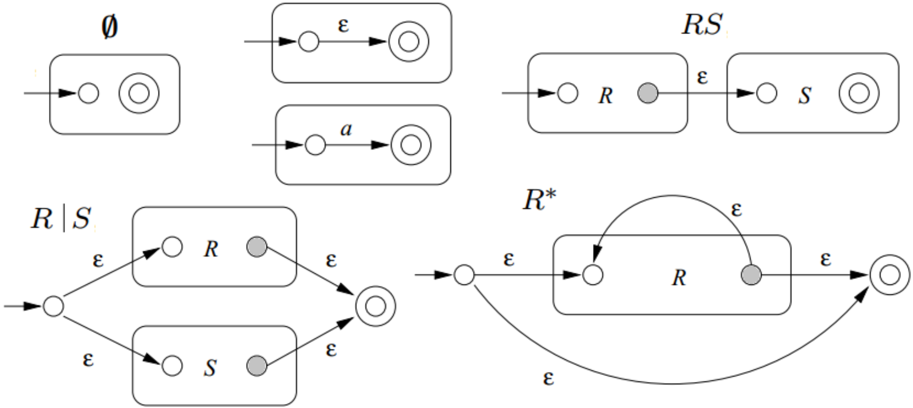
Método de Thompson



http://users.encs.concordia.ca/~grahne/hmu_slides/main.pdf

Linguagens Regulares → Autômatos

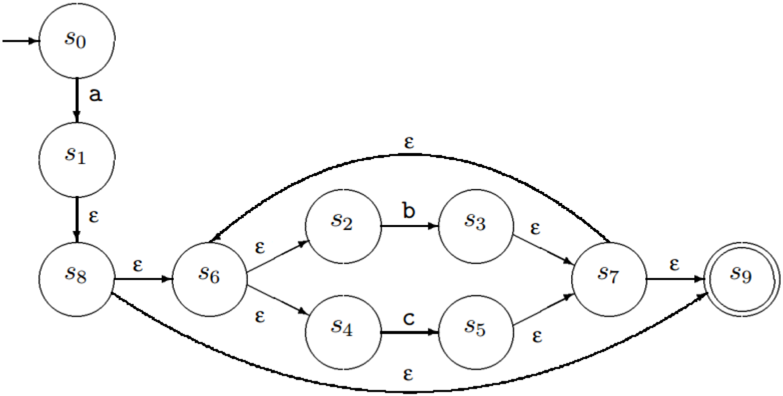
Método de Thompson **Exemplo: a (b | c)***



http://users.encs.concordia.ca/~grahne/hmu_slides/main.pdf

Linguagens Regulares → Autômatos

Método de Thompson **Exemplo:** **a (b | c)***



Sumário

Algumas propriedades das linguagens regulares

- Se R é uma linguagem regular então $R^c = \Sigma^* - R$ é regular

  Prova: Construa um AFD para R e troque os estados finais por não finais e viceversa.

  Exemplo: A linguagem $((0|1)*01(0|1)*)^c$ é regular?

- Se R é uma linguagem regular então R^r também é regular

  Prova: Construa um AFD para R; troque a direção de todas as transições; troque todos os estados finais por não finais; faça o antigo estado inicial ser final; coloque um novo estado inicial com ε-transições até todos os antigos estados finais.

  Exemplo: A linguagem $((0|1)*01(0|1)*)^r$ é regular?

Universidade Federal do ABC

Algumas propriedades das linguagens regulares

- Se R_1 e R_2 são linguagens regulares sobre Σ então $R_1 \cap R_2$ também é uma linguagem regular sobre Σ.

  Prova: $R_1 \cap R_2 = (R_1^c \cup R_2^c)^c$. De forma alternativa, partindo dos AFDs para R_1 e R_2, construa $A_{R_1 \cap R_2}$ como segue:

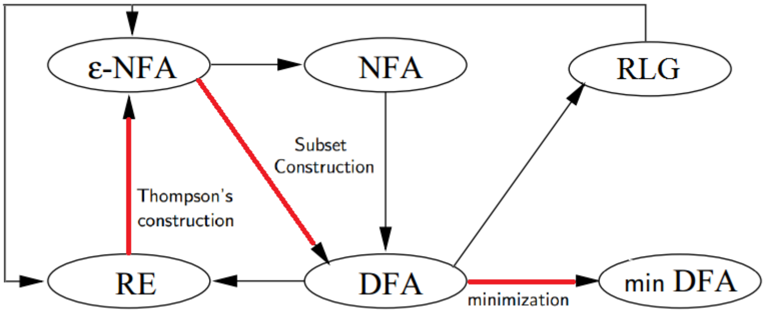
  $$A_{R_1 \cap R_2} = (Q_1 \times Q_2, \Sigma, \delta, (q_{01}, q_{02}), F_1 \times F_2)$$

  onde $\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$

  Exemplo: $\Sigma^* b \Sigma^* \cap \Sigma^* c \Sigma^*$ com $\Sigma = \{a, b, c, d\}$ é regular?

Expressões Regulares ↔ Autômatos↔ **REG**

- **REG** classe das linguagens regulares, amplamente estudada
- Poder expressivo suficiente para um grande # de aplicações
- Algoritmos de reconhecimento com tempo lineal w.r.t. o tamanho da entrada



RE: regular expressions; DFA (NFA): (non-) deterministic automata; RLG: right-linear grammars

Algumas Ferramentas

	AFD/AFND	GR	ER	AP	MT	Saída Gráfica	Correção Automática
GAM	X					X	
VAS	X				X	X	
Language Emulator	X	X	X				
SCTMF	X	X	X	X	X		
JFLAP	X	X	X	X	X	X	
LabLF	X	X	X	X	X	X	X

Tabela 1: Ferramentas e seus modelos de computação implementados.

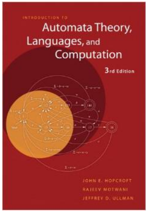
http://www.br-ie.org/pub/index.php/rbie/article/viewFile/1208/1108

https://regex101.com/r/vS7vZ3/224#javascript

https://regexr.com/

Sumário

Bibliografia Básica



1. **Introduction to Automata Theory, Languages, and Computation** (3rd Edition). J. Hopcroft, R. Motwani and J. Ullman. Addison-Wesley, 2006

   `http://www-db.stanford.edu/~ullman/ialc.html`

   `http://users.encs.concordia.ca/~grahne/hmu_slides/`

2. Introduction to the Theory of Computation. M. Sipser
3. **Compilers: Principles, Techniques, and Tools** (2nd Edition). Alfred Aho, Monica Lam, Ravi Sethi, and Jeffrey Ullman. Addison-Wesley, 2006
4. Linguagens Formais e Autômatos, P. B. Menezes