



UFABC

Apresentação do Projeto Dirigido

**Um sistema para auxiliar na
aprendizagem da disciplina
Linguagens Formais e Autômatos**

Rafael Cardoso da Silva
21048012

08/11/2016

MCTA015-13 - Linguagens Formais e Autômatos (3-1-4)

8º quadrimestre

MCTA004-13 - Arquitetura de Computadores	MCTA002-13 - Algoritmos e Estruturas de Dados II	MCTA027-13 - Teoria dos Grafos	MCTA005-13 - Banco de Dados	MCTA014-13 - Inteligência Artificial
--	---	-----------------------------------	--------------------------------	--------------------------------------

9º quadrimestre

MCTA022-13 - Redes de Computadores	MCTA026-13 - Sistemas Operacionais	MCTA015-13 - Linguagens Formais e Autômatos	MCTA010-13 - Engenharia de <i>Software</i>	BCS0002-15 - Projeto Dirigido	Livre
--	--	---	--	----------------------------------	-------

10º quadrimestre

MCTA019-13 - Projeto de Graduação em Computação I	MCTA025-13 - Sistemas Distribuídos	MCTA007-13 - Compiladores	MCTA016-13 - Paradigmas de Programação	Opção limitada
--	--	------------------------------	--	----------------

Fonte: bcc.ufabc.edu.br/grade-curricular/estrutura-da-grade.html

Ementa:

- Conceitos básicos.
- Linguagens regulares:
 - Autômatos determinísticos
 - Autômatos não-determinísticos
 - Expressões regulares.
- Linguagens livres de contexto:
 - Gramática
 - Autômatos a pilha.
- Linguagens recursivamente enumeráveis:
 - Máquinas de Turing determinísticas e não-determinísticas.
- ...

Definido por: **$M = (Q, \Sigma, \delta, s, F)$**

- conjunto não vazio de estados Q ;
- um alfabeto Σ ;
- um estado inicial $s \in Q$;
- um conjunto de estados de aceitação $F \subseteq Q$;
- uma função de transição $\delta: Q \times \Sigma \rightarrow Q$.

$$\hat{\delta}: Q \times \Sigma^* \rightarrow Q$$

- i.* $\hat{\delta}(q, \varepsilon) = q$ para todo estado $q \in Q$;
- ii.* $\hat{\delta}(q, \sigma) = \delta(q, \sigma)$ para todo estado $q \in Q$ e símbolo $\sigma \in \Sigma$;
- iii.* $\hat{\delta}(q, \sigma_1 \cdot \dots \cdot \sigma_k) = \delta(\hat{\delta}(q, \sigma_1 \cdot \dots \cdot \sigma_{k-1}), \sigma_k)$ para todo estado $q \in Q$ e símbolos $\sigma_1, \dots, \sigma_k \in \Sigma$.

- A **linguagem reconhecida** por um autômato é o conjunto de palavras que são aceitas por esse autômato, ou seja, é definida pelo conjunto:

$$L(\text{autômato}) = \{w \in \Sigma^* : \hat{\delta}(s, w) \in F\}$$

Exemplo M_1

$L(M_1) = \{w \in \Sigma^* : w \text{ tem número par de símbolos } a\}$.

- $Q = \{S_1, S_2\}$

- $\Sigma = \{a, b\}$ entrada

- $s = S_1$

- $F = \{S_1\}$

- | δ | a | b |
|----------|-------|-------|
| S_1 | S_2 | S_1 |
| S_2 | S_1 | S_2 |

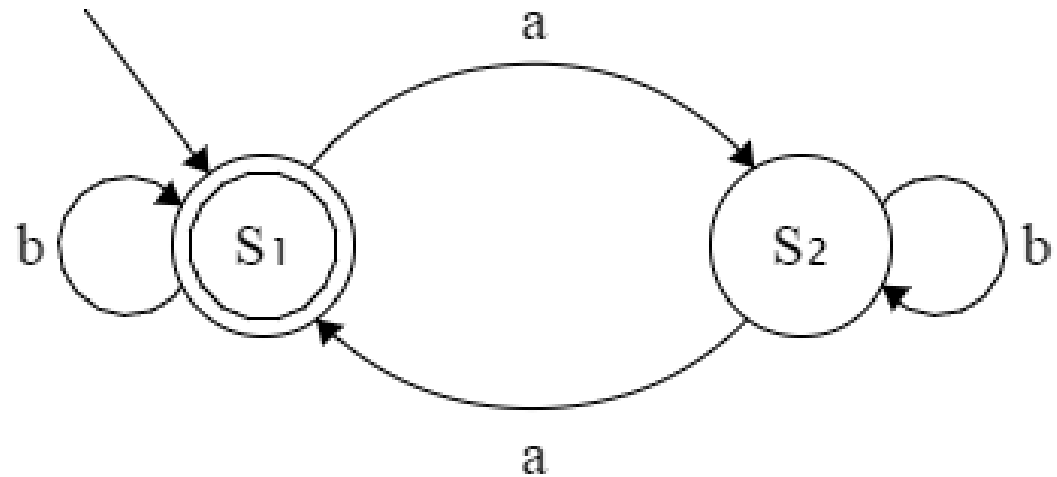


Figura 1: Autômato M_1 .

Outro Exemplo

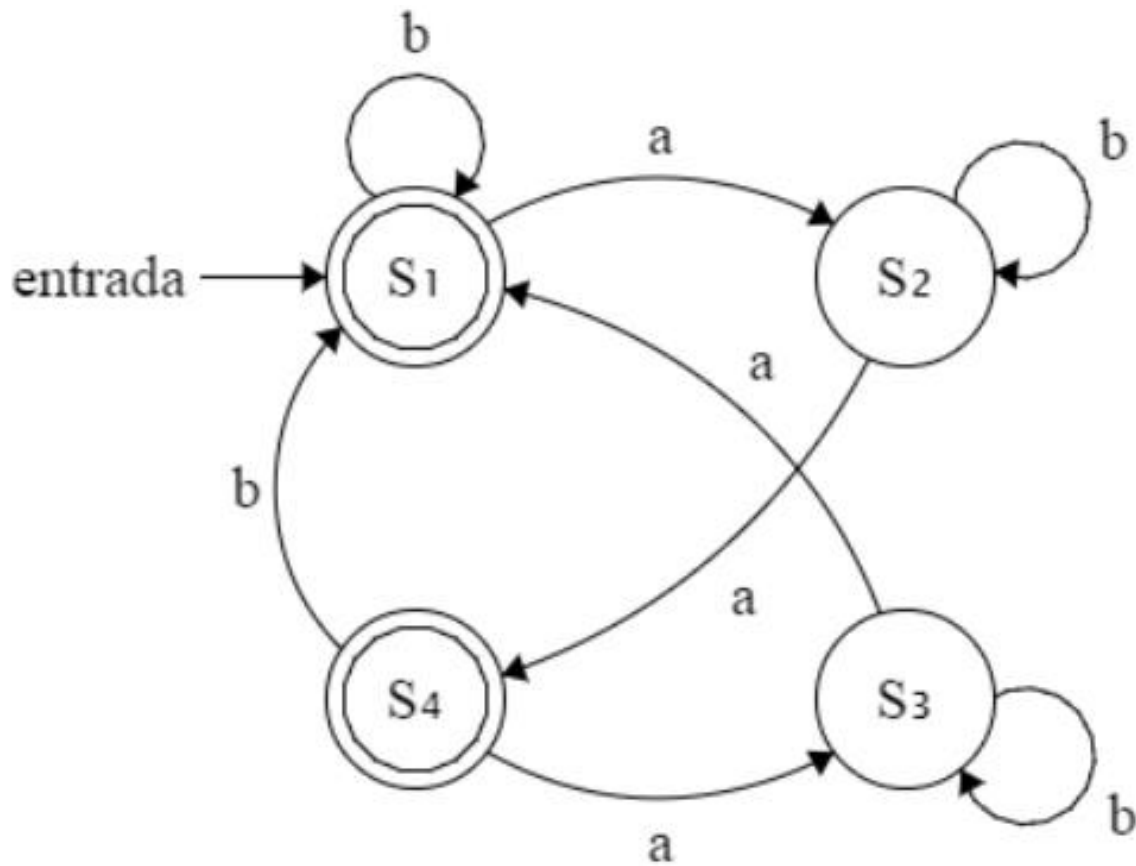


Figura 2: Autômato M_2 .

São Equivalentes!

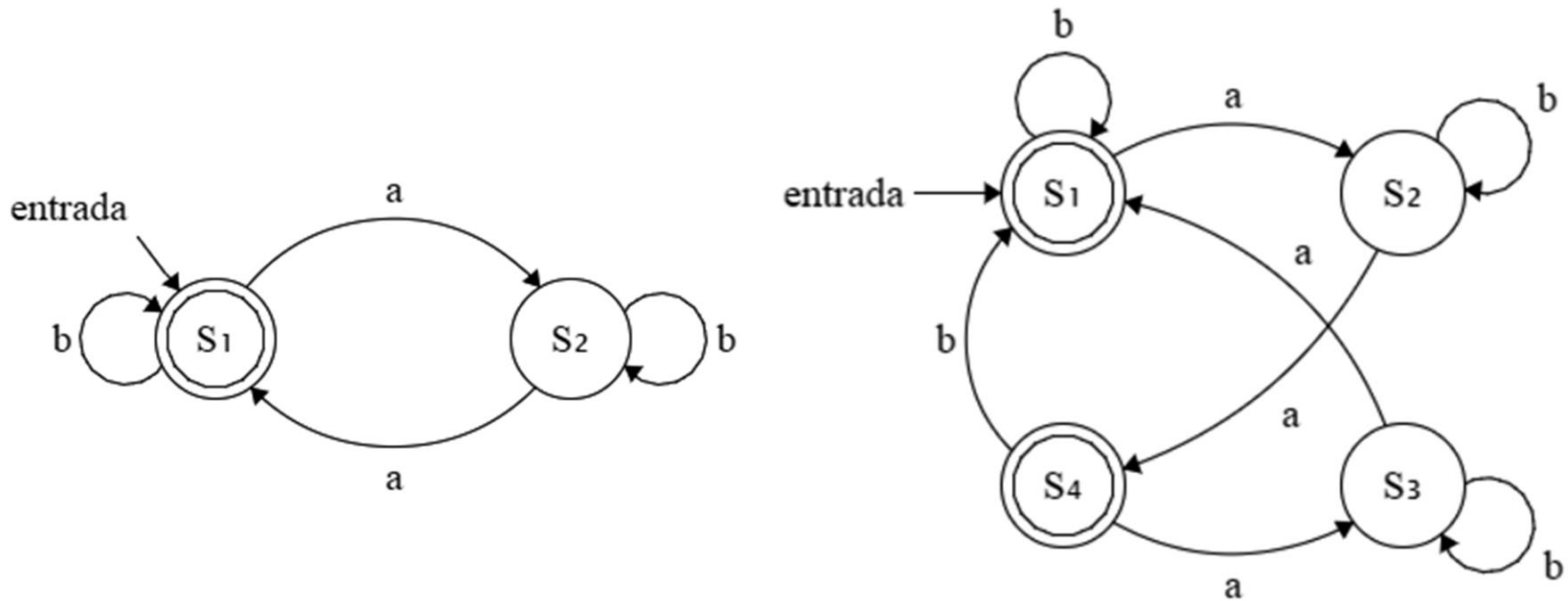


Figura 3: Autômato M_1 a esquerda e Autômato M_2 a direita.

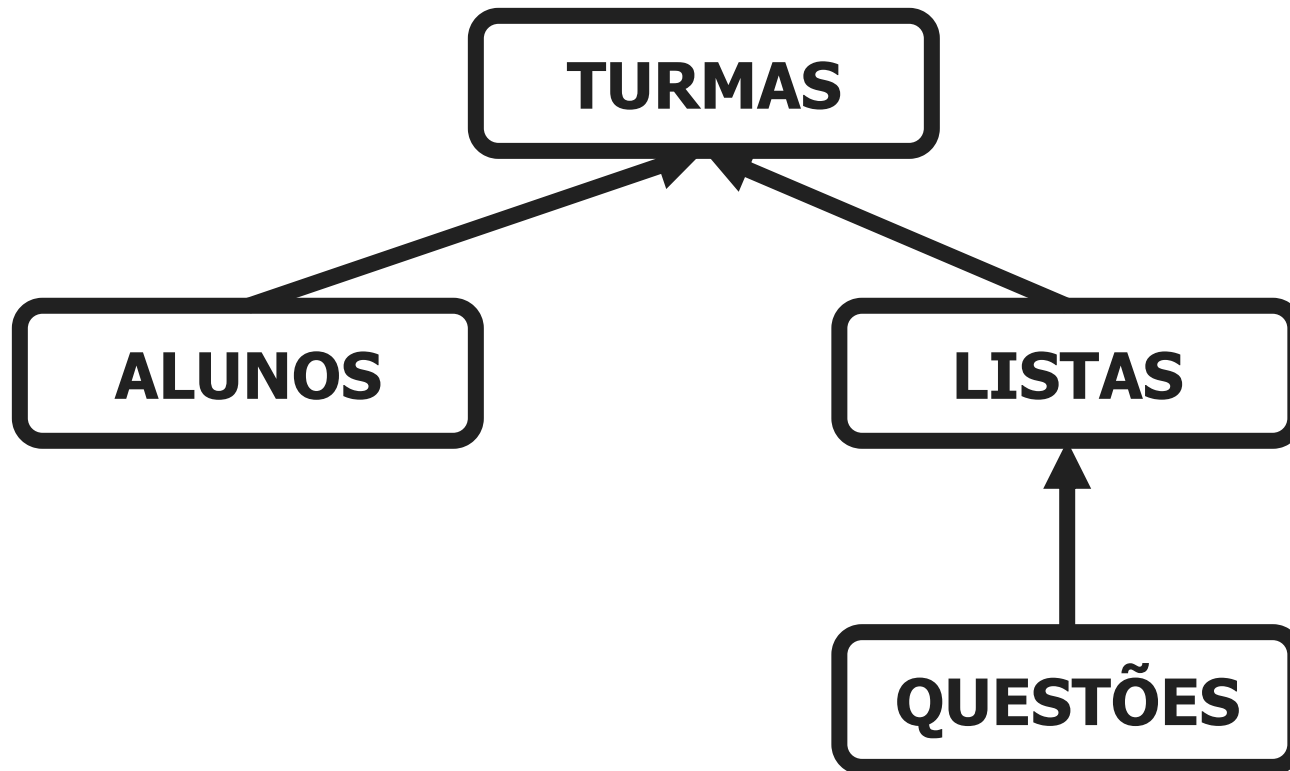
Decidir se dois autômatos finitos determinísticos reconhecem a mesma linguagem.

- É um problema bem resolvido!
- dois estados q_1 e q_2 são equivalentes se, para toda palavra $w \in \Sigma^*$ vale:

$$\hat{\delta}(q_1, w) \in F \iff \hat{\delta}(q_2, w) \in F$$

- Esse problema está intimamente relacionado com o problema de minimização de AFDs
- Os primeiros algoritmos desenvolvidos:
 - **(HUFFMAN, 1954)**
 - **(MOORE, 1956)**
- Complexidade $O(n^2)$

- O algoritmo mais eficiente conhecido para minimização de autômatos executa em tempo $O(n \log n)$ (**HOPCROFT, 1971**).
- Posteriormente Hopcroft e Karp desenvolveram um algoritmo linear para testar a equivalência de autômatos (**HOPCROFT; KARP, 1971**).



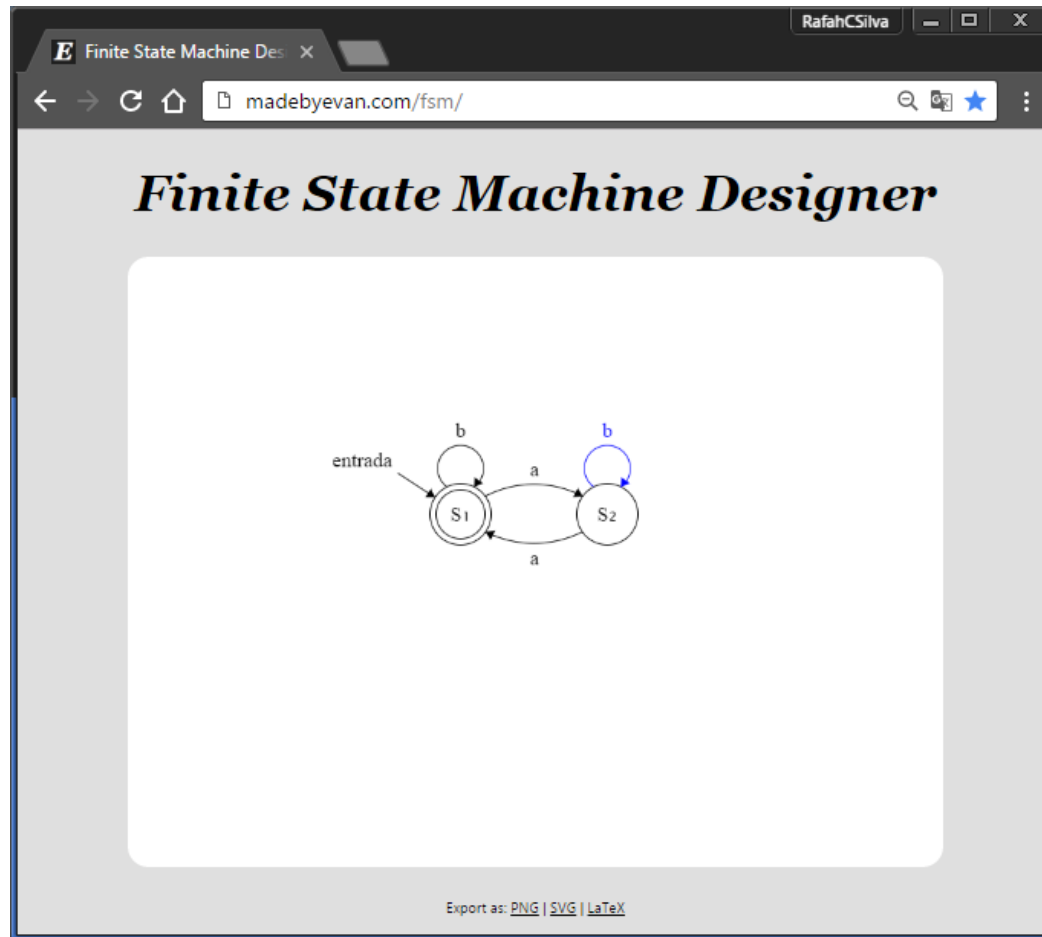


Figura 4: FSMdesigner desenvolvido por Evan Wallace.

- Correto
- Incorreto
 - Palavra que os distinguiram
- Salvo com Rascunho

Objetivos e Metodologia

- Estudar e implementar o algoritmo de Moore para minimização e equivalência de AFDs (MOORE, 1956);
- Estudar e implementar o algoritmo de Hopcroft e Karp para testar a equivalência de AFDs (HOPCROFT; KARP, 1971);
- Desenvolvimento do sistema de apoio a aprendizagem da disciplina Linguagens Formais e Autômatos;
- Reuniões semanais com o orientador, afim de acompanhar o progresso do desenvolvimento do sistema e esclarecer eventuais dúvidas.

1. Leitura da bibliografia (capítulos de livros e artigos citados).
2. Implementação dos algoritmos em C++.
3. Adequação do FSM Designer às necessidades deste projeto.
4. Elaboração do relatório parcial.
5. Desenvolvimento do sistema.
 - 5.1. Levantamento de requisitos, análise e projeto do sistema.
 - 5.2. Implementação do sistema.
6. Confeção dos exercícios no sistema.
7. Aplicação numa turma experimental.
8. Elaboração do relatório final.

Tabela 1: Cronograma.

	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul
1	x	x	x	x	x	x	x					
2	x	x	x	x								
3				x	x							
4					x	x	x					
5						x	x	x	x	x		
5.1						x						
5.2							x	x	x	x		
6									x	x		
7										x	x	x
8									x	x	x	x

Tabela 2: Orçamento para a execução deste projeto.

CUSTEIO	
Serviços de Terceiro	
(5 meses) Hospedagem do Site	R\$ 74,50 (R\$ 14,90* cada mês)
(1 ano) Registro de Domínio	R\$ 49,90*
TOTAL	R\$ 124,40

* valores obtidos com base nos pacotes oferecidos no site da LocaWeb¹ .

¹ Acessado dia 29/10/2016, disponível em <<http://www.locaweb.com.br/hospedagem-de-sites/>>

SIPSER, M. Introduction to the Theory of Computation. [S.l.]: Cengage Learning, 2012.

HUFFMAN, D. A. The synthesis of sequential switching circuits. I, II. J. Franklin Inst., v. 257, p. 161–190, 275–303, 1954. ISSN 0016-0032.

MOORE, E. F. Gedanken-experiments on sequential machines. In: Automata studies. [S.l.]: Princeton University Press, Princeton, N. J., 1956, (Annals of mathematics studies, no. 34). p. 129–153.

HOPCROFT, J. An $n \log n$ algorithm for minimizing states in a finite automaton. In: Theory of machines and computations (Proc. Internat. Sympos., Technion, Haifa, 1971). [S.l.]: Academic Press, New York, 1971. p. 189–196.

HOPCROFT, J. E.; KARP, R. M. A linear time algorithm for testing equivalence of finite automata. Technical report of Cornell University, p. 71–114, 1971.

WALLACE, E. Finite State Machine Designer. 2010. <<http://madebyevan.com/fsm/>>. [Online; acessado 29 de outubro de 2016].



UFABC

OBRIGADO PELA ATENÇÃO