

## Estudo de Linguagens Formais: Conceitos e Prática Aplicados na Construção de um Semáforo

**Raimundo José Macário Costa,**      **José Luiz dos Anjos Rosa**

Depto. de Ciências de Computação e Matemática, CCMAT, UEZO

23070-200, Rio de Janeiro, RJ

E-mail: mac\_costa@yahoo.com, jrosa@uezo.rj.gov.br

***Resumo:** Esse trabalho apresenta a construção de um semáforo de trânsito, onde pedestres solicitam que o semáforo libere a passagem. Esse trabalho foi desenvolvido no decorrer da disciplina Linguagens Formais e Autômatos, na qual são apresentados os conceitos formais e fundamentais, tais como, os sistemas autômatos e máquinas de estado.*

### **Introdução:**

A área da computação tornou-se, em pouco mais de 60 anos, um verdadeiro universo para a área científica. Segundo Ramos (2009), o profissional moderno pode atuar em subáreas diversificadas, as quais requerem conhecimentos especializados e aptidões variadas. E, assim, o aprendizado dos fundamentos da computação pode significar a garantia de formação de futuros profissionais melhor preparados para a área, além de permitir o desenvolvimento de tecnologias inovadoras e novos produtos.

De acordo com o estudo de Ramos (2009), o conhecimento contido na área de linguagens formais e autômatos é derivado de duas subáreas de conhecimento, consideradas independentes, uma da outra, até a década de 1960: por um lado linguagens formais, e por outro a teoria de autômatos. Foi a partir de pesquisas efetuadas nessa época que se estabeleceu a relação entre as duas subáreas, e a partir de então foram combinadas numa única grande área de conhecimento, sendo consideradas, hoje, indissociáveis uma da outra.

A teoria de linguagens formais trata da caracterização, classificação, formalização e estudo das propriedades das linguagens estruturadas, em frases. Por sua vez a teoria dos autômatos é o estudo de computadores abstratos, também chamados de máquinas de estado, tendo surgido em 1930 a partir das pesquisas de Alan Turing. O objetivo de Turing era descrever, com exatidão, o que a máquina seria capaz de processar. Após o surgimento dos estudos da máquina de Turing, diversos outros cientistas e pesquisadores trouxeram suas contribuições, porém, a mais expressiva na área foi a contribuição de Noam Chomsky, em meados de 1950, apresentando uma classificação das linguagens estruturadas em frases, organizadas em níveis de complexidade crescente, tornando-se referência fundamental para o estudo das linguagens formais. A partir de então, passou a ser conhecida como Hierarquia Chomsky (Hopcroft, Motwani e Ullman, 2002).

A Hierarquia Chomsky contempla as classes: das linguagens regulares, ou tipo 3; das linguagens livre de contexto, ou tipo 2; das linguagens sensíveis ao contexto, ou tipo 1; e, por fim, das linguagens recursivas enumeráveis, ou tipo 0 (Hopcroft, Motwani e Ullman, 2002).

Se, por um lado, a subárea das linguagens formais oferece uma perspectiva de estudo baseada na síntese de cadeias para o estudo das várias classes de linguagens, por outro lado, a teoria de autômatos oferece uma perspectiva de estudo baseada na análise de cadeias com o mesmo objetivo.

Os autômatos nos dias de hoje estão dotados de funções específicas de controle e canais de comunicação que permitem ligá-los entre si e a computadores em rede, formando um sistema integrado. Os autômatos são divididos em autômatos finitos e autômatos à pilha. Porém, neste trabalho será utilizado somente o conceito de autômatos finitos. Este trabalho tem por objetivo, apresentar a construção de um semáforo utilizando os conceitos de linguagens formais e autômatos finitos discutidos na disciplina de Introdução à Teoria de Autômatos, Linguagens e Computação. Este trabalho foi desenvolvido por dois alunos durante o segundo semestre de 2013, na disciplina Linguagens Formais, Autômatos e Computação do curso de Ciência da

Computação no Centro Universitário Estadual da Zona Oeste – UEZO, na cidade do Rio de Janeiro.

## Autômatos finitos

Autômatos finitos, ou máquinas de estados finitos, são uma forma matemática de descrever tipos particulares de algoritmos. Autômatos finitos podem ser utilizados para descrever o processo de reconhecimento de padrões em cadeias de entrada.

Os autômatos finitos são basicamente grafos, tais como os diagramas de transição, com algumas diferenças: 1) os autômatos finitos são reconhecedores; eles simplesmente dizem “sim”, ou “não”, sobre cada possível cadeia de caracteres de entrada; 2) os autômatos finitos podem ser de dois tipos (Aho *et al*, 2008):

- a) Autômatos finitos não determinísticos (NFA – do inglês: Nondeterministic Finite Automata) não têm restrições sobre os rótulos de suas arestas. Um símbolo pode rotular várias arestas saindo do mesmo estado, e sendo  $\epsilon$ , a cadeia vazia, um rótulo possível.
- b) Autômatos finitos determinísticos (DFA – do inglês: Deterministic Finite Automata) possuem, para cada estado e para cada símbolo de seu alfabeto de entrada, exatamente uma aresta com esses símbolos saindo desse estado.

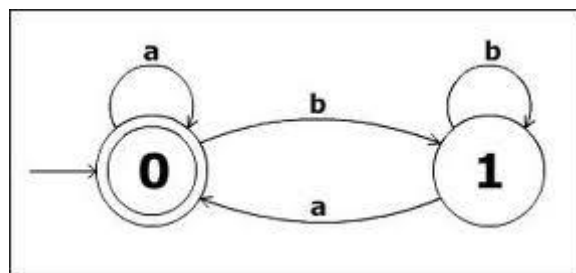


Figura 1: modelo de autômatos

Um DFA é representado por uma quintupla:  $(Q, \Sigma, \delta, q_0, F)$

Onde:

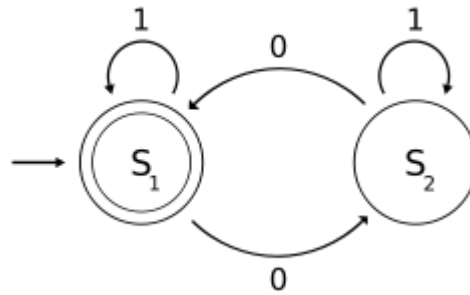
- um conjunto finito de estados ( $Q$ )
- um conjunto finito de símbolos de entrada chamado Alfabeto ( $\Sigma$ )
- uma função de transição ( $\delta : Q \times \Sigma \rightarrow Q$ )
- um estado inicial ( $q_0 \in Q$ ) e
- um conjunto de estados de aceitação ( $F \subseteq Q$ )

Seja  $w = a_1 a_2 \dots a_n$  uma cadeia de símbolos sobre o alfabeto  $\Sigma$ , O autômato  $M$  aceita a cadeia  $w$  se somente se existe uma sequência de estados,  $r_0, r_1, \dots, r_n$ , em  $Q$  com as seguintes condições:

$$\begin{aligned}
 r_0 &= q_0 \\
 r_{i+1} &= \delta(r_i, a_{i+1}), \text{ para } i = 0, \dots, n-1 \\
 r_n &\in F.
 \end{aligned}$$

Em outras palavras, a primeira condição afirma que a máquina se inicia no estado inicial  $q_0$ . A segunda condição diz que, dado cada símbolo da entrada  $w$ , a máquina transita de estado em estado de acordo com a função de transição  $\delta$ . A terceira e última condição diz que a máquina aceita  $w$  se somente se o último símbolo da entrada leva o autômato a parar em um estado  $f$  tal que  $f \in F$ . Caso contrário, diz-se que a máquina rejeita a entrada. O conjunto de

cadeias que  $M$  aceita é chamado Linguagem reconhecida por  $M$  e é simbolicamente representado por  $L(M)$ .



Os autômatos finitos determinísticos e não determinísticos são capazes de reconhecer as mesmas linguagens.

## Máquinas de estado

Uma máquina de estado finito é uma modelagem de um comportamento composto por estados, transições e ações.

Um estado armazena informações sobre o passado; uma transição indica uma mudança de estado e é descrita por uma condição que precisa ser realizada para que a transição ocorra; uma ação é a descrição de uma atividade que deve ser realizada em determinado momento.

A máquina de estado é representada por um diagrama bastante simplificado, conhecido como diagrama de transição de estado, que tem como objetivo facilitar o entendimento de qualquer pessoa interessada no processo.

## Experimento: Descrição do Semáforo

Foi desenvolvido um semáforo para servir como sistema de controle de trânsito, possuindo três estados: Aberto (indicado pela cor verde), Atenção (indicado pela cor amarela) e Fechado (indicado pela cor vermelha). Quando o semáforo está sinalizado em “Aberto”, indica que a passagem está permitida para carros; quando está sinalizado em “Atenção”, indica que os carros devem reduzir a velocidade e, posteriormente, parar o carro, e, quando está em “Fechado”, indica que a passagem está aberta para pedestres.

O semáforo troca de estado em instantes de tempo, sendo 10 segundos o tempo de troca do estado “Aberto” para “Atenção”, três segundos para trocar do estado “Atenção” para “Fechado”, e novamente 10 segundos para trocar do estado “Fechado” para “Aberto”.

Além da troca de estados pelo tempo determinado, o pedestre pode solicitar que o semáforo mude para o estado “Fechado”. Com isso, se o pedestre solicitar a mudança de estado quando o semáforo estiver em “Fechado” o semáforo acrescenta em quatro segundos do tempo de transição de estado. Se o pedestre solicitar a mudança de estado com o semáforo em “Atenção”, nada acontece, e se solicitar com o semáforo em “Aberto” o sistema decrementa quatro segundos no tempo de transição de estados (caso tenha menos de quatro segundos o sistema muda de estado).

O sistema só permite uma solicitação por estado. Após atender a solicitação o semáforo volta aos tempos estabelecidos de transição de estado.

## Máquina de estado do semáforo

Estado do semáforo: Aberto, Atenção e Fechado.

Relações (transições) de estado: acontece de dois modos: pelo tempo pré-definido ou por solicitação do pedestre.

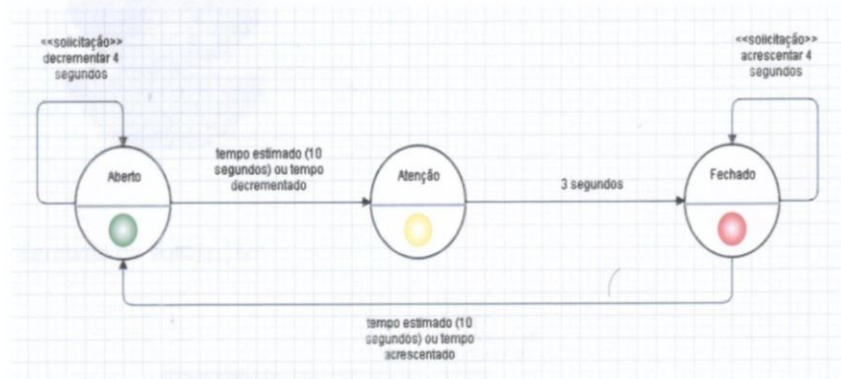


Figura 2: Máquina de estado do semáforo

## Software Auger

Auger é uma ferramenta para apoiar o ensino de autômatos finitos na área de Ciência da Computação. O autômato é construído por meio de um editor gráfico. Elaborado o autômato, o usuário poderá executar e acompanhar, passo a passo o processo de minimização para chegar ao autômato finito determinístico mínimo e simular o reconhecimento de sentenças (Szymanski e Garcindo, 2004).

Neste trabalho, foi utilizada a ferramenta Auger para realizar a simulação do autômato. Esta ferramenta permite criar a máquina de estados, simular/testar o andamento da máquina de estados e gerar a matriz de transição de estados.

Para o entendimento da máquina de estados criada na ferramenta, é necessário seguir os seguintes passos ou legendas:

- Q0: semáforo aberto (verde) – Estado inicial.
- Q1: semáforo atenção (amarelo).
- Q2: semáforo fechado (vermelho) – Estado final.
- X: tempo estimado (10 segundos ou acréscimo e decréscimo da solicitação).
- Y: tempo estimado (3 segundos)
- S: solicitação feita.

## Software simulador de autômatos

Como a ferramenta Auger não gera a gramática através do autômato criado optou-se por utilizar o software Simulador de autômatos para gerar a gramática, como pode ser observado na imagem da tabela gerada.

Utilize ? para indicar transição vazia			
Produções			
1	S	-> XA	(q1.X) = q2
2	S	-> SS	(q1.S) = q1
3	A	-> YB	(q2.Y) = q3
4	B	-> XS	(q3.X) = q1
5	B	-> SB	(q3.S) = q3
6	B	-> ?	(Estado de aceitação)

Figura 4: Tabela criada pelo Simulador de Autômatos

## Desenvolvimento do autômato: Linguagem escolhida

Para esta etapa, foi utilizada a ferramenta Arduino, placa de controle I/O (Entrada e Saída) baseada no microcontrolador ATmega (Atmel), e que serve de controle para diversos outros sistemas.

Arduino é uma plataforma *open-source* de desenvolvimento de projetos digitais baseada em hardware (placa microcontroladora) e software de código aberto (ambiente de desenvolvimento para escrever comandos para a placa), permitindo o recebimento de sinais de entrada para controle de dispositivos diversos.

O microcontrolador do Arduino é programado utilizando-se a linguagem de programação Arduino (baseado no Wiring - *framework open-source* de programação para microcontroladores). Seu ambiente de desenvolvimento é baseado no Processing, linguagem de programação baseada em Java.

### Imagens do sistema



Figura 4: Semáforo de automóveis

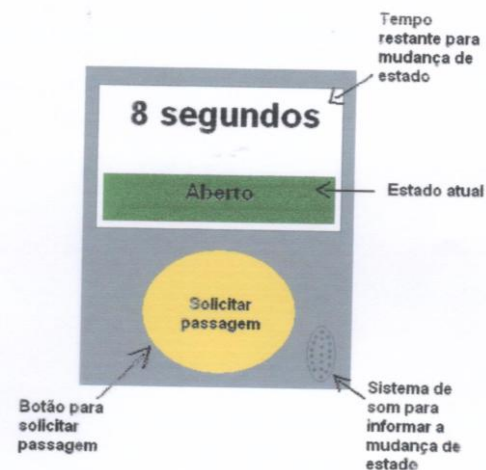


Figura 5: Terminal de solicitação

## Conclusão

O desenvolvimento do semáforo apresentado consegue mostrar, na prática, a aplicação dos conceitos formais apreendidos, tais como autômatos finitos, máquinas de estado e ferramentas que agregam conhecimentos voltados para a disciplina Linguagens Formais e Autômatos.

O entendimento dos conceitos utilizados pelos alunos foi motivador no âmbito da disciplina ainda que em caráter introdutório, constituindo elemento incentivador para o estudo, servindo, ainda, para uma maior contextualização do conhecimento clássico com as pesquisas de ponta na área. O uso de ferramentas de experimentação prática utilizada durante a aula permitiu romper a aridez algébrica do raciocínio lógico e do pensamento abstrato, permitindo que os alunos fossem estimulados em suas próprias formas de aprendizado.

Procurou-se, porém, detectar se a introdução de elementos concretos (Arduino) e de conhecimento de mundo dos estudantes (sinais de controle trânsito) resultaria em ganhos de aprendizagem da matéria alvo da pesquisa por parte dos alunos.

Pelos resultados obtidos, houve pequeno ganho na aprendizagem da matéria (transição para estado superior de conhecimento, no jargão próprio do assunto em estudo), muito embora, segundo os alunos, o domínio dos conceitos sobre máquinas de estado e grafos seja preponderante para a construção de protótipos eficientes e mais evoluídos. Ou seja, o uso de ferramentas de experimentação facilita os estudos iniciais da matéria Linguagens Formais e Autômatos, mas deve ser utilizada como complemento às atividades derivadas da abstração em sala de aula.

O uso de simuladores abertos, em contraposição aos fechados, oferece a vantagem de criar pontos adicionais de contato com outras disciplinas da área de computação, o que contribuiu de forma decisiva para a motivação, retenção e aprendizado dos alunos durante o desenvolvimento da disciplina.

Foi demonstrado, de forma prática, que a construção de um terminal de solicitação para pedestres é algo amplamente viável para o sistema de trânsito, o que acarreta um tempo de espera reduzido, evitando acidentes.

## **Agradecimentos**

Os autores agradecem à FAPERJ o apoio financeiro na realização deste projeto.

## **Referências**

[3] Aho, et al. *Compiladores - Princípios, Técnicas e Ferramentas*. 2. Ed. – São Paulo: Pearson Addison-Wesley, 2008.

[3] Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D. *Introdução À Teoria de Autômatos, Linguagens e Computação*. Editora: Campus. 2002.

[1] Ramos, Marcus Vinícius Midena. Ensino de linguagens formais e autômatos em cursos superiores de computação. *ReCeT - Revista de Computação e Tecnologia da PUC-SP*. Vol. 1, Nº1, 2009.

[2] Szymanski, Charbel (Universidade do Sul de Santa Catarina) e Garcindo, Luiz (UNISUL). Auger - Ferramenta de apoio ao ensino de autômatos finitos. *WORKCOMPSUL*, 2004. Acesso em: 24 de fevereiro, 2014. Disponível: <http://inf.unisul.br/~ines/workcomp/cd/pdfs/2952.pdf>.