UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

ENGENHARIA ELETRÔNICA

THAIS YURIKO DE GROOT MIDORIKAWA

**SIMULADOR DE MÁQUINA DE ESTADOS EM C**

Atividade Prática Supervisionada

Curitiba

2017

THAIS YURIKO DE GROOT MIDORIKAWA

**SIMULADOR DE MÁQUINA DE ESTADOS EM C**

Atividade prática supervisionada apresentada como requisito parcial para aprovação na disciplina de Estrutura de Dados 1, do curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Bruno Sens Chang.

Curitiba

2017

**SUMÁRIO**

[1 INTRODUÇÃO 4](#_Toc484109448)

[1.1 MOTIVAÇÃO 4](#_Toc484109449)

[2 DESCRIÇÃO DO PROBLEMA 4](#_Toc484109450)

[2.1 RELAÇÃO COM A MATÉRIA 6](#_Toc484109451)

**REFERÊNCIAS 7**

1. INTRODUÇÃO

Máquinas de estados finitos consistem em, basicamente, representações dos estados de um problema e como estes estão relacionados entre si a partir de ações. Assim, tais máquinas podem ser utilizadas para se modelar problemas lógicos complexos de forma a dividir as tarefas a serem resolvidas em problemas menores de uma maneira sistemática. Estas máquinas variam quanto aos tipos de saída gerada, podendo ser aceitadoras ou transdutoras, além de poderem ser distinguidas quanto à quantidade de transições para cada entrada, sendo classificadas como determinísticas ou não-determinísticas.

As máquinas de estados finitos determinísticas, ou autômatos finitos determinísticos (AFD), consistem em máquinas que possuam apenas um estado correspondente a uma determinada entrada, sendo os autômatos finitos não-determinísticas (AFN) o oposto. Apesar de possuir uma complexidade de inferior aos AFNs, os AFDs podem ser aplicados em inúmeros problemas de diferentes áreas, cobrindo a maior parte das necessidades cotidianas que possam vir a necessitar o uso destas máquinas.

* 1. MOTIVAÇÃO

Visto isso, este estudo tem como motivação principal a sua utilização posterior em projetos pessoais envolvendo eletrônica e microcontroladores, além de experiências com o conceito de máquinas de estados em disciplinas cursadas anteriormente. Assim, objetiva a implementação de um simulador não-gráfico de máquinas de estados finitos em linguagem C.

1. DESCRIÇÃO DO PROBLEMA

O conceito de máquinas de estados finitos é bastante antigo e popular, havendo hoje diversas implementações de simuladores destas máquinas em diferentes linguagens de programação e com formas diferentes de construção destas máquinas, como pode ser visto em [1-4].

Em [1], a implementação visual da máquina é obtida a partir de uma expressão regular inserida pelo usuário, de forma que os estados podem ser percorridos dado uma entrada pelo usuário. Já em [2], o usuário manipula graficamente os elementos da página a fim de se montar um diagrama de blocos correspondente à máquina desejada, podendo inicia-la ou percorrê-la passo-a-passo. Por sua vez, o simulador de [3] consiste em um arquivo executável, no qual o usuário, assim como em [2], deve utilizar-se de elementos gráficos para criar a máquina desejada, porém sua representação visual é a partir de grafos. Já [4] consiste no código fonte de um simulador de máquinas de estados finitos na linguagem Python. Por fim, [5] e [6] apresentam simuladores de AFD em C, nos quais o usuário fornece ao programa todos os dados necessários para a criação da AFD e, no fim, este pode inserir uma sequência de símbolos e obter o estado resultante desta sequência.

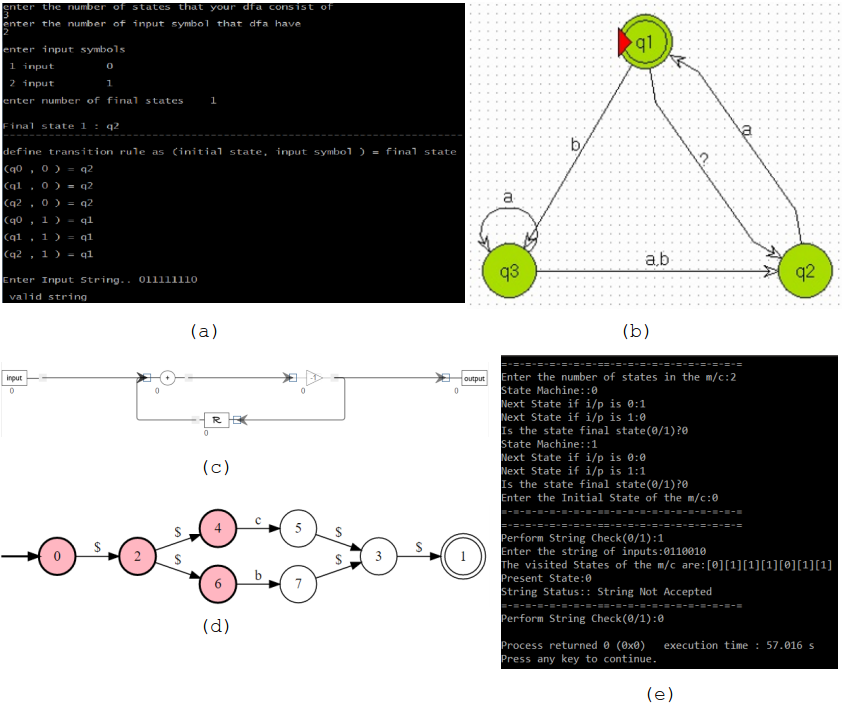


Figura 1. (a) Tela do programa em [6]; (b) Grafo do programa em [3]; (c) Diagrama de blocos de [2]; (d) Grafos a partir de expressão regular em [1]; (e) Tela do programa em [5].

Assim, o simulador a ser implementado neste projeto visa cobrir um determinado número de estados a fim de, além da elaboração de um simulador de AFD, pôr em prática os conhecimentos aprendidos ao longo do curso. O usuário deverá inicialmente definir a quantidade de estados de sua máquina e as entradas correspondentes de cada estado, bem como o estado inicial da máquina. Assim, para cada entrada (previamente registrada), o AFD deverá informar ao usuário em qual estado se encontra até que lhe seja fornecido um critério de parada.

* 1. RELAÇÃO COM A DISCIPLINA

Os conceitos aprendidos em sala que serão explorados no decorrer deste trabalho compreendem listas encadeadas, recursões e grafos.

**REFERÊNCIAS**

[1] <http://ivanzuzak.info/noam/webapps/fsm_simulator/>

[2] <http://web.mit.edu/6.mitx/www/6.01-statemachine/mvc.html>

[3] <http://www.simuladordeautomatos.com/>

[4] <https://gist.github.com/hoodwink73/11165659>

[5] <https://wowmoron.wordpress.com/2015/02/14/simulating-a-dfa-in-c/>

[6] <http://www.cprogramto.com/c-program-for-dfa/>