

Sumário

INTRODUÇÃO	4
OBJETIVOS.....	5
AUTÔMATO FINITO DETERMINÍSTICO.....	6
EXEMPLO DE MINIMIZAÇÃO.....	7
CONCLUSÃO	11
REFERÊNCIAS	12

INTRODUÇÃO

O Autômato Finito Determinístico (AFD) é um assunto abordado na teoria dos autômatos, presente na disciplina de Linguagens Formais Autômatos, que consiste em uma máquina composta por uma fita, uma unidade de controle e transições, possuindo o poder de aceitar ou rejeitar uma determinada cadeia de símbolos.

Dado um AFD, é possível realizar sua minimização, sendo reduzido o número de estados utilizados na versão original, mas mantendo a equivalência, pois possuirão a mesma linguagem regular. Se dois estados são equivalentes, então poderão ser unidos em um só estado. Mas para isso, deve-se levar em consideração que o autômato seja determinístico, não pode ter estados inacessíveis e a função de transição deve ser total.

Este trabalho apresenta a implementação dessa minimização, respeitando os requisitos necessários.

OBJETIVOS

Desenvolver um algoritmo que realize a minimização de um Autômato Finito Determinístico (AFD) dado pelo usuário.

Realizar a minimização sem alterar a linguagem do AFD original.

Compreender o funcionamento da minimização de AFD.

Praticar a lógica de programação aplicada a Linguagens Formais Autômatos.

AUTÔMATO FINITO DETERMINÍSTICO

O Autômato Finito Determinístico (AFD) é um modelo computacional que possui estados, tendo obrigatoriamente, um estado inicial e um ou mais estados finais, transições, e uma unidade de controle para indicar qual o estado está sendo processado no momento. Com isso, ele pode aceitar ou rejeitar palavras.

Dentre sua descrição formal, podemos encontrar:

- Alfabeto (Σ): são os caracteres que existem na palavra dada.
- Conjunto de estados (E): todos os estados utilizados para o AFD.
- Estado inicial (i): como o próprio nome diz, o estado em que se dá início ao autômato.
- Estados finais (F): estados que podem encerrar o autômato.
- Tabela de transição: movimentações entre entradas.

EXEMPLO DE MINIMIZAÇÃO

Dado o AFD abaixo:

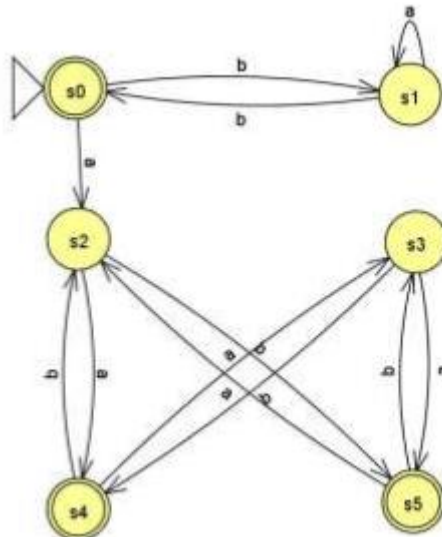
$\Sigma = \{a,b\}$

$E = \{s0, s1, s2, s3, s4, s5\}$

$i = s0$

$F = \{s0, s4, s5\}$

	a	b
s0	s1	s2
s1	s1	s0
s2	s4	s5
s3	s5	s4
s4	s3	s2
s5	s2	s3



O processo de minimização será dado por:

```

4
5
6 Digite a quantidade de estados:
6
7 Digite o Estado Inicial:
0
8 Digite a quantidade de letras do alfabeto:
2
9 Digite a 0 letra do alfabeto:
a
10 Digite a 1 letra do alfabeto:
b
11 Digite a quantidade de estados finais:
3
12 Digite o 0 estado final:
0
13 Digite o 1 estado final:
4
14 Digite o 2 estado final:
5
15
16 *****
17 *          Tabela guia!          *
18 *****
19
20      a      b
21 $0      0,0      0,1
22 $1      1,0      1,1
23 $2      2,0      2,1
24 $3      3,0      3,1
25 $4      4,0      4,1
26 $5      5,0      5,1
27 Digite o estado para 0,0 :
1
28 Digite o estado para 0,1 :
2
29 Digite o estado para 1,0 :
1
30 Digite o estado para 1,1 :
0
31 Digite o estado para 2,0 :
4
32 Digite o estado para 2,1 :
5
33 Digite o estado para 3,0 :
5
34 Digite o estado para 3,1 :
4
35 Digite o estado para 4,0 :
3
36 Digite o estado para 4,1 :
2
37 Digite o estado para 5,0 :
2
38 Digite o estado para 5,1 :
3

```

```

*****
*          Verificacao alcancavel!          *
*****
$0:true
$1:true
$2:true
$3:true
$4:true
$5:true
*****
*   Tabela Preenchida!   *
*****
      a      b
$0      $1      $2
$1      $1      $0
$2      $4      $5
$3      $5      $4
$4      $3      $2
$5      $2      $3
*****
*   Tabela Triangular!   *
*****
$1      1,0
$2      2,0      2,1
$3      3,0      3,1      3,2
$4      4,0      4,1      4,2      4,3
$5      5,0      5,1      5,2      5,3      5,4
$0      $1      $2      $3      $4
*****
* Pares Trivialmente nao equivalentes *
*****
$1      X
$2      X      null
$3      X      null      null
$4      null      X      X      X
$5      null      X      X      X      null
$0      $1      $2      $3      $4
Anulando os campos: 2 1
Anulando os campos: 3 1
Anulando os campos: 4 0
Anulando os campos: 5 0
*****
* Pares nao marcados *
*****
$1      X
$2      X      X
$3      X      X      null
$4      X      X      X      X
$5      X      X      X      X      null
$0      $1      $2      $3      $4

```

```

*****
*   Juntando estados   *
*****
Criando um novo estado!
Criando um novo estado!
*****
*   Tabela minimizada!   *
*****
      a      b
$0      $1      $6
$1      $1      $0
$6      $7      $7
$7      $6      $6
*****
*   Lista de estados!   *
*****
$0
$1
$6
$7
*****
*   Lista de estados finais!   *
*****
$0
$7
*****
*   Estado Inicial!   *
*****
$0
*****
*   Alfabeto!   *
*****
a
b
*****
Digite X para sair...
=

```

DECISÕES DE PROJETO PARA A IMPLEMENTAÇÃO

Na implementação foi utilizada o conceito de programação orientada a objetos, utilizando a linguagem Java e o compilador Eclipse Java Oxygen. Para o armazenamento dos dados inseridos pelo usuário foram criadas variáveis estáticas de diferentes tipos, entre elas: int, boolean, char, String e até mesmo foi criada uma classe chamada Estado para uma melhor manipulação e entendimento do algoritmo.

Para a organização desses dados, foram utilizados arrays simples e bidimensionais, utilizando como auxílio uma lista. As limitações desse algoritmo estão nas matrizes nas quais já foram pré-estipuladas o tamanho máximo de 999 por 999, impossibilitando assim minimizações de autômatos muito grandes.

Usando conceitos aprendidos na matéria de Inteligência Artificial e com a ajuda do monitor de LFA foi implementado uma busca por profundidade para verificar se o autômato tinha todos os estados alcançáveis. O estado de erro também foi implementado, mas esse usando ideias mais simples e estruturadas. A impossibilidade de inserção de mais de uma transição com a mesma letra por estado faz com que todos os pré-requisitos para a minimização sejam verificados, fazendo com que o algoritmo trabalhe de forma mais consistente e segura.

CONCLUSÃO

A minimização do AFD tem como característica a redução de estados utilizados originalmente, mas mantendo a equivalência. Sendo assim, o autômato pode realizar menos transições, pois a quantidade de estados será otimizada ao máximo. Porém, para poder minimizar de maneira correta, os requisitos apresentados devem ser levados em conta.

A implementação do trabalho ajuda a entender melhor as etapas que o autômato deve tomar a fim de realizar a minimização completa, além de ser possível determinar se dado AFD, ele poderá ser minimizado ou não.

Com isso, foi possível compreender melhor os estudos aplicados ao AFD, conforme vimos durante o decorrer das aulas teóricas.

REFERÊNCIAS

BROOKSHEAR, J.G. **Ciência da computação**. Porto Alegre: Bookman, 2000.

DELAMARO, Márcio Eduardo. **Linguagens Formais e Autômatos**. Maringá: UEM, 1998;

MENEZES, P. B.; DIVERIO, T. A. **Teoria da Computação: Máquinas Universais e Computabilidade**. 3ª edição Bookman 2011.