



# UNIFEI

UNIVERSIDADE FEDERAL DE ITAJUBÁ,  
CAMPUS ITABIRA

ECO113 - TEORIA DA COMPUTAÇÃO

---

## Trabalho Prático 1

---

***Professor:*** Rafael Francisco dos Santos

***Alunos:***

Felipe Wallace Pinto Coelho - 2018011779

Thiago Luis Freitas Carvalho - 2018011723

17 de junho de 2023

## 1 Introdução

O presente relatório descreve a implementação de um Autômato Finito Determinístico (AFD) em linguagem C++. Neste trabalho, foi desenvolvida uma implementação capaz de ler um arquivo de entrada contendo as informações necessárias para definir o autômato. O arquivo de entrada contém as seguintes informações:

- Lista de estados: Os estados que compõem o AFD.
- Alfabeto: O conjunto de símbolos que podem ser utilizados como entrada.
- Transições: As regras que definem a função de transição do autômato.
- Estado inicial: O estado a partir do qual o autômato inicia a leitura das palavras.
- Estados finais: O conjunto de estados que representam o estado de aceitação do autômato.
- Palavras de teste: Um conjunto de palavras a serem testadas pelo autômato.

## 2 Desenvolvimento

A implementação lê o arquivo de entrada e armazena as informações em estruturas de dados adequadas, como listas e mapas. Em seguida, realiza a simulação do autômato para cada palavra de teste, percorrendo as transições e atualizando o estado atual conforme os caracteres da palavra. Ao final, verifica se o estado atual está presente na lista de estados finais, determinando se a palavra é aceita ou não pelo autômato.

- Exemplo do arquivo de entrada de dados:

```
1 Primeira linha: os N estados;  
2 Segunda linha: os M simbolos do alfabeto;  
3 As proximas N linhas: sao compostas por M valores indicando as  
  transicoes de cada um dos estados;  
4 A proxima linha: tem o estado inicial;  
5 A proxima linha: tem F estados finais;  
6 A proxima linha: tem um numero T qualquer indicando o numero de  
  palavras que serao testadas;  
7 As proximas T linhas: contem 1 palavra cada, que devem ser  
  validas no automato.
```

Listing 1: Exemplo de arquivo de entrada

## 2.1 Resultados de testes realizados

Todas as palavras do alfabeto  $\Sigma = a, b, c$  que tenha o sufixo *aca* ou *bca*.

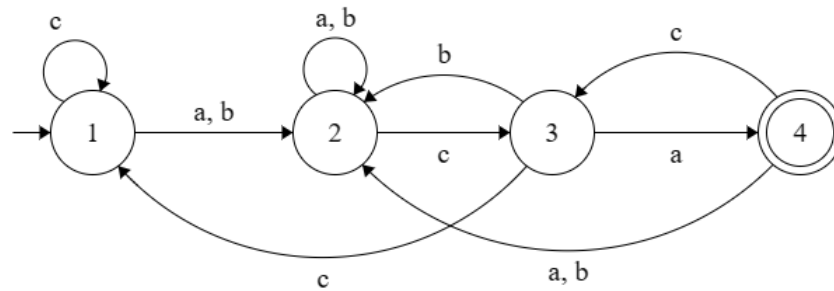


Figura 1: AFD 1

```

1 1 2 3 4
2 a b c
3 2 2 1
4 2 2 3
5 4 2 1
6 2 2 3
7 1
8 4
9 5
10 aaaca
11 acbca
12 cccbca
13 abcabc
14 bbbaaa
  
```

Listing 2: Exemplo de arquivo de entrada para a AFD 1

```

>_ Console x Shell x +
> make -s
> ./main
A palavra 'aaaca' pertence a linguagem
A palavra 'acbca' pertence a linguagem
A palavra 'cccbca' pertence a linguagem
A palavra 'abcabc' não pertence a linguagem
A palavra 'bbbaaa' não pertence a linguagem
>
  
```

Figura 2: Saída no terminal para o AFD 1

Conjunto das cadeias sobre  $a, b$  que contém um número par de substrings  $ba$ .

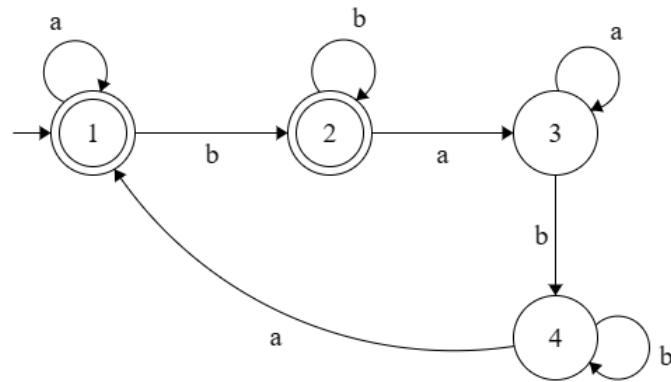


Figura 3: AFD 2

```

1 1 2 3 4
2 a b
3 2 4
4 1 3
5 4 2
6 3 1
7 1
8 2
9 5
10 abb
11 ababa
12 baa
13 bababaaba
14 a
  
```

Listing 3: Exemplo de arquivo de entrada para a AFD 2

```

>_ Console x Shell x +
❖ make -s
❖ ./main
A palavra 'abb' pertence a linguagem
A palavra 'ababa' pertence a linguagem
A palavra 'baa' não pertence a linguagem
A palavra 'bababaaba' pertence a linguagem
A palavra 'a' pertence a linguagem
❖ 
  
```

Figura 4: Saída no terminal para o AFD 2

Conjunto das cadeias sobre  $a, b$  que contém a substring  $aabba$ .

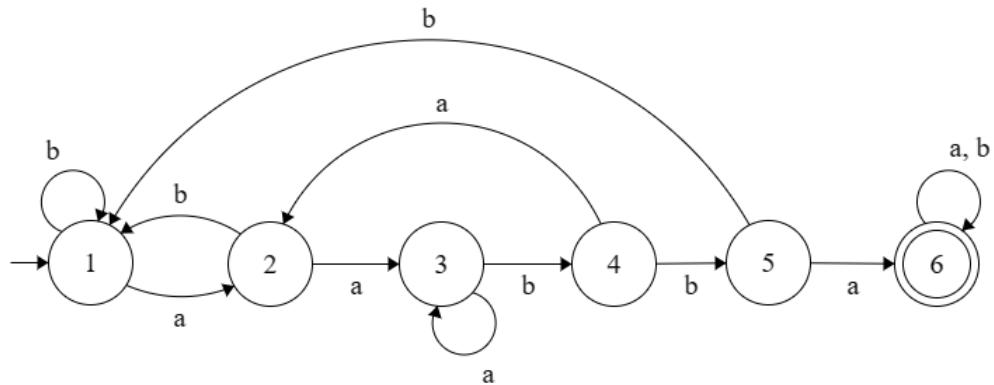


Figura 5: AFD 3

```

1 1 2 3 4 5 6
2 a b
3 2 1
4 3 1
5 3 4
6 2 5
7 6 1
8 6 6
9 1
10 6
11 5
12 aabba
13 aaabbab
14 aabbabababbb
15 aabbba
16 bbaa

```

Listing 4: Exemplo de arquivo de entrada para a AFD 3

```

>_ Console x Shell x +
> make -s
> ./main
A palavra 'aabba' pertence a linguagem
A palavra 'aaabbab' pertence a linguagem
A palavra 'aabbabababbb' pertence a linguagem
A palavra 'aabbba' não pertence a linguagem
A palavra 'bbaa' não pertence a linguagem
>

```

Figura 6: Saída no terminal para o AFD 3

Conjunto das cadeias sobre  $a, b$  com um número ímpar de  $a$ 's e um número par de  $b$ 's.

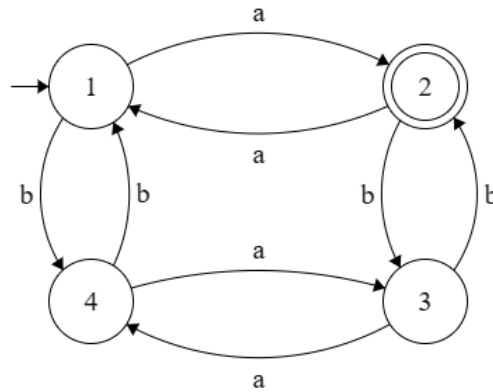


Figura 7: AFD 4

```

1 1 2 3 4
2 a b
3 1 2
4 3 2
5 3 4
6 1 4
7 1
8 1 2
9 5
10 bbbaba
11 baba
12 aabb
13 aaba
14 babaab
    
```

Listing 5: Exemplo de arquivo de entrada para a AFD 4

```

>_ Console x Shell x +
> make -s
> ./main
A palavra 'bbbaba' pertence a linguagem
A palavra 'baba' pertence a linguagem
A palavra 'aabb' pertence a linguagem
A palavra 'aaba' não pertence a linguagem
A palavra 'babaab' pertence a linguagem
>
    
```

Figura 8: Saída no terminal para o AFD 4

Exemplo retirado do PDF do trabalho.

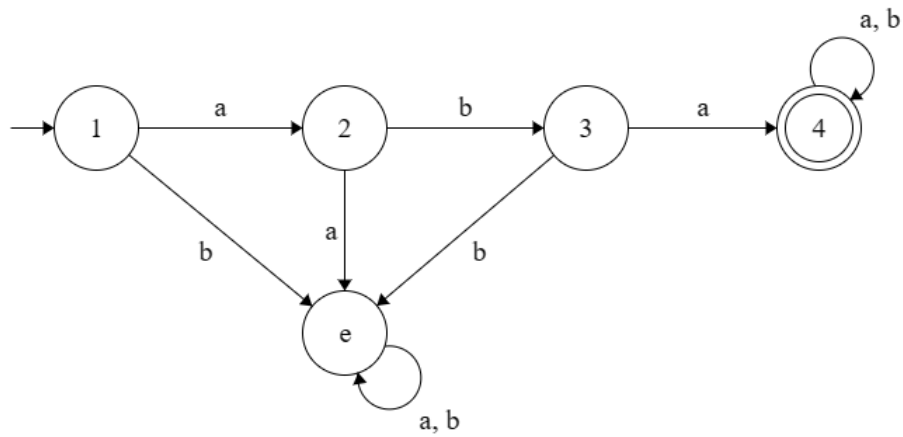


Figura 9: AFD 5

```
1 1 2 3 4 e
2 a b
3 2 e
4 e 3
5 4 e
6 4 4
7 e e
8 1
9 4
10 5
11 abaaa
12 abbababa
13 abaabaaba
14 aa
15 abaabbbb
```

Listing 6: Exemplo de arquivo de entrada para a AFD 5

```
>_ Console x Shell x +
> make -s
> ./main
A palavra 'abaaa' pertence a linguagem
A palavra 'abbababa' não pertence a linguagem
A palavra 'abaabaaba' pertence a linguagem
A palavra 'aa' não pertence a linguagem
A palavra 'abaabbbb' pertence a linguagem
>
```

Figura 10: Saída no terminal para o AFD 5

### 3 Conclusão

Os resultados obtidos com a implementação do AFD foram satisfatórios, demonstrando a correta classificação das palavras de teste conforme a linguagem definida pelo autômato. A implementação foi capaz de lidar com diferentes casos de teste e apresentou um bom desempenho.

Conclui-se que a implementação de autômatos finitos determinísticos em linguagem de programação é uma forma eficiente de validar teorias aprendidas na disciplina de Teoria da Computação. O trabalho permitiu a aplicação prática dos conhecimentos adquiridos e proporcionou uma maior compreensão sobre o funcionamento dos autômatos finitos determinísticos.