

## Manual do Usuário - Simulador Universal de Autômatos Finitos

### 1. Introdução

Este manual fornece instruções detalhadas sobre como utilizar o simulador universal de autômatos finitos (AFDs e AFNs) implementado em Python. O simulador permite a análise de linguagens regulares através da simulação de autômatos finitos, permitindo que usuários especifiquem a estrutura do autômato e testem seu comportamento em diferentes cadeias de entrada.

### 2. Requisitos

Para utilizar o simulador, você precisará dos seguintes requisitos:

- **Sistema Operacional:** O simulador foi testado e funciona em diversos sistemas operacionais Windows.
- **Python:** O simulador foi desenvolvido em Python 3.x. Certifique-se de ter a versão correta do Python instalada em seu sistema. Pode ser necessário verificar a documentação do simulador para a versão específica recomendada.
- **Arquivos do Simulador:** Você precisará baixar os seguintes arquivos:
  - `main.py`: Código fonte do programa.
  - `simulate.exe` (Windows) ou outro executável dependendo do sistema operacional: Arquivo executável do programa.
  - `doc/manual.pdf`: Manual do Usuário (este arquivo).

### 3. Instruções de Uso

1. **Baixe os arquivos:** Baixe os arquivos `main.py`, `simulate.exe` (ou equivalente para o seu sistema operacional) e `doc/manual.pdf` para o mesmo diretório no seu computador.
2. **Crie os arquivos de entrada e saída:**
  - **Arquivo de entrada (`input.txt`):** Crie um arquivo de texto chamado `input.txt` que contenha a especificação do autômato e as cadeias de entrada a serem simuladas. O formato do arquivo de entrada é o seguinte:
    1. **Número de estados:** A primeira linha do arquivo deve conter o número de estados do autômato, assumindo nomes de `q0` a `qn-1`, onde `n` é o número de estados.
    2. **Conjunto de símbolos terminais:** A segunda linha do arquivo deve conter o conjunto de símbolos terminais, separados por espaços.
    3. **Conjunto de estados de aceitação:** A terceira linha do arquivo deve conter o conjunto de estados de aceitação, separados por espaços.
    4. **Número de transições:** A quarta linha do arquivo deve conter o número de transições do autômato.
    5. **Transições:** As linhas seguintes do arquivo devem conter as transições do autômato, no formato `q x q' s`, onde `q` e `q'` são estados, `s` é um símbolo terminal ou `-` (lambda) e `x` é a ação.
    6. **Número de cadeias de entrada:** A linha após as transições deve conter o número de cadeias de entrada a serem simuladas.
    7. **Cadeias de entrada:** As linhas seguintes do arquivo devem conter as cadeias de entrada a serem simuladas, uma em cada linha.

◦ **Exemplo de arquivo de entrada (`input.txt`):**

```
3
a b
1 2
6
0 a 1
0 b 1
1 a 1
1 b 2
2 a 0
2 b 2
10
abbbba
aabbbb
bbabbabbabb
bbbbbbbbbbbbbb
-
ababababababab
bbbaabbbb
abba
a
aaa
```

\* **Arquivo de saída (`output.txt`)**: O programa gerará um arquivo de saída chamado `output.txt` que conterá os resultados da simulação. Cada linha do arquivo de saída conterá uma cadeia de entrada e o resultado da simulação (aceita ou rejeita).

3. **Execute o simulador:** Abra um prompt de comando (cmd) e navegue até o diretório onde os arquivos estão armazenados. Execute o seguinte comando:

```
python main.py input.txt output.txt # Uso recomendado (pode variar
dependendo da distribuição)
```

- Substitua `main.py`, `input.txt` e `output.txt` pelos nomes dos seus arquivos.

4. **Analise os resultados:** Abra o arquivo `output.txt` para visualizar os resultados da simulação. Cada linha do arquivo conterá uma cadeia de entrada e o resultado da simulação (aceita ou rejeita).

### Observações

- O número máximo de estados é 10 (pode ser um valor ajustável).
- O número máximo de símbolos terminais é 10 (pode ser um valor ajustável).
- O número máximo de transições é 50 (pode ser um valor ajustável).
- O comprimento máximo de cada cadeia de entrada é 20 símbolos (pode ser um valor ajustável).

- O programa verifica se os dados de entrada estão no formato correto antes de iniciar a simulação.
- Em caso de erros ou problemas, consulte o manual de uso ou entre em contato com o desenvolvedor.

## 5. Exemplos de Uso

O manual do usuário deve incluir exemplos de como especificar AFDs e AFNs em arquivos de entrada e como interpretar os resultados da simulação.

- **Exemplo 1: Simular um AFD que reconhece a linguagem regular  $(a+b)a^*bb^*(a(a+b)a^*)b$**

- Arquivo de entrada (`input.txt`):

```
3
a b
2 3
6
0 a 1
0 b 1
1 a 1
1 b 2
2 a 3
2 b 2
10
abbbba
aabbbb
bbabbabbabb
bbbbbbbbbbbbbb
-
ababababababab
bbbbaabbbb
abba
a
aaa
```

- Arquivo de saída (`output.txt`):

```
abbbba - Aceita
aabbbb - Aceita
bbabbabbabb - Rejeita
bbbbbbbbbbbbbb - Aceita
- - Rejeita (cadeia vazia)
ababababababab - Rejeita
bbbbaabbbb - Aceita
abba - Aceita
a - Rejeita
aaa - Aceita
```

- **Exemplo 2: Simular um AFN que reconhece a linguagem regular  $a^*(b|c)^*$**

- Arquivo de entrada (**input.txt**): (Modifique o arquivo para refletir o AFN)

```
... (definição do AFN)
...
```

- Arquivo de saída (**output.txt**): (O programa gerará resultados similares ao exemplo 1)

## 6. Limitações

Mencione quaisquer limitações conhecidas do simulador, como o número máximo de estados, símbolos ou cadeias de entrada. Indique se o código-fonte pode ser modificado para aumentar esses limites.

## 7. Considerações Finais

O manual do usuário pode incluir um breve resumo dos benefícios de usar o simulador para análise de linguagens regulares. Incentive o usuário a explorar o simulador para diferentes cenários e sugerir possíveis aplicações.

## 8. Apêndice

Você pode incluir um apêndice com formalismos matemáticos relacionados a AFDs e AFNs, se o público-alvo do manual se beneficiar desse conhecimento adicional.

## 9. Glossário

Este glossário define alguns termos técnicos usados neste manual relacionados a autômatos finitos:

- **Autômato Finito (AF):** Um modelo matemático que reconhece linguagens regulares. Ele consiste de um conjunto finito de estados, um conjunto de símbolos terminais, uma função de transição e um conjunto de estados de aceitação.
- **Estado:** Uma configuração possível do autômato.
- **Símbolo Terminal:** Um símbolo da linguagem que o autômato pode processar.
- **Função de Transição:** Define o próximo estado do autômato com base no estado atual e no símbolo lido.
- **Estado de Aceitação:** Um estado que indica que a cadeia de entrada processada é reconhecida pela linguagem.
- **Linguagem Regular:** Um conjunto de cadeias finitas formadas por um conjunto finito de símbolos, de acordo com um conjunto de regras.
- **Autômato Finito Determinístico (AFD):** Um autômato finito onde a função de transição sempre leva a um único estado seguinte para um determinado estado e símbolo.
- **Autômato Finito Não Determinístico (AFN):** Um autômato finito onde a função de transição pode levar a vários estados possíveis para um determinado estado e símbolo.

## 10. Referências

- Introduction to Automata Theory, Languages, and Computation (3rd Edition) by John E. Hopcroft e Jeffrey D. Ullman (<https://www.amazon.com/Introduction-Automata-Theory-Languages-Computation/dp/0321455363>)
- [https://en.wikipedia.org/wiki/Automata\\_theory](https://en.wikipedia.org/wiki/Automata_theory)

