

Relatório do Simulador de Autômatos

Rafael Souza Tavares de Castro-8179

Guilherme Francisco de Sousa Costa-7555

Introdução

Este relatório descreve o funcionamento de um simulador de autômatos capaz de simular Autômatos Finitos Determinísticos (AFD), Autômatos Finitos Não Determinísticos (AFN) e Máquinas de Turing. O simulador foi desenvolvido em Python e permite a inserção de autômatos no sistema, a verificação de palavras e a conversão de AFNs para AFDs. Além disso, o sistema permite a visualização de seus estados, transições, e conjuntos de estados de aceitação por meio da impressão no terminal.

Funcionalidades Principais

1. Simulação de Autômatos Finitos Determinísticos (AFD):

- Permite a inserção de estados, alfabeto, transições, estado inicial e estados de aceitação.
- Permite a verificação se uma palavra é aceita pelo autômato ou não.
- Possui uma função de impressão no terminal

2. Simulação de Autômatos Finitos Não Determinísticos (AFN):

- Permite a inserção dos mesmos componentes que o AFD, mas com transições não determinísticas.
- Realiza a conversão de um AFN para AFD.
- Também pode verificar a aceitação de palavras tanto no AFN original quanto no AFD convertido.
- Possui uma função de impressão no terminal

3. Simulação de Máquinas de Turing:

- Permite a criação de uma máquina com estados, alfabeto de entrada, alfabeto da fita, símbolo de marcação de início, símbolo de espaço vazio, transições, estado inicial e estados de aceitação.

- Capaz de computar palavras fornecidas pelo usuário, modificando a fita conforme as transições definidas.
- A simulação termina quando a máquina alcança um estado de aceitação ou quando o número máximo de passos é atingido.
- Possui uma função de impressão no terminal

Exemplos

Exemplo 1 - AFN:

- **Descrição:** Este AFN reconhece a linguagem de todas as palavras que contêm pelo menos um 'a'.
- **Definição:**
 - Estados: $\{q_0, q_1\}$
 - Alfabeto: $\{a, b\}$
 - Transições:
 - $q_0 \xrightarrow{a} q_1$
 - $q_0 \xrightarrow{b} q_0$
 - $q_1 \xrightarrow{a} q_1$
 - $q_1 \xrightarrow{b} q_1$
 - Estado inicial: q_0
 - Estados de aceitação: $\{q_1\}$

Este autômato não determinístico aceita palavras como "a", "ba", "bba", mas rejeita palavras como "bbb".

Exemplo 2 - AFN:

- **Descrição:** Este AFN reconhece a linguagem de todas as palavras que terminam com o símbolo 'b'.
- **Definição:**
 - Estados: $\{q_0, q_1\}$
 - Alfabeto: $\{a, b\}$
 - Transições:

$q_0 \xrightarrow{a} q_0$

$q_0 \xrightarrow{b} q_1$

$q_1 \xrightarrow{a} q_0$

$q_1 \xrightarrow{b} q_1$

- Estado inicial: q_0
- Estados de aceitação: $\{q_1\}$

Este autômato aceita palavras como "b", "ab", "aab", mas rejeita palavras como "a", "aa".

Exemplo 3 - AFD:

- **Descrição:** Este AFD reconhece a linguagem de palavras binárias que contêm um número par de '0's.
- **Definição:**
 - Estados: $\{q_0, q_1\}$
 - Alfabeto: $\{0, 1\}$
 - Transições:
 - $q_0 \xrightarrow{0} q_1$
 - $q_0 \xrightarrow{1} q_0$
 - $q_1 \xrightarrow{0} q_0$
 - $q_1 \xrightarrow{1} q_1$
 - Estado inicial: q_0
 - Estados de aceitação: $\{q_0\}$

Este AFD aceita palavras como "110", "1001", mas rejeita palavras como "101" ou "11".

Exemplo - Máquina de Turing

A Máquina de Turing implementada no simulador é capaz de incrementar números binários. Ela varre a fita da direita para a esquerda, aplicando as transições adequadas para realizar o incremento, como em um contador binário.

- **Definição:**

- Estados: $\{q_0, q_1, q_f\}$
- Alfabeto de entrada: $\{0, 1\}$
- Alfabeto da fita: $\{0, 1, x, -\}$
- Marca de Início: $-$
- Vazio: x
- Transições:
 - $(q_0, 1) \rightarrow (q_0, 1, >)$
 - $(q_0, 0) \rightarrow (q_0, 0, >)$
 - $(q_0, x) \rightarrow (q_1, x, <)$
 - $(q_1, 1) \rightarrow (q_1, 0, <)$
 - $(q_1, 0) \rightarrow (q_f, 1, >)$
 - $(q_1, -) \rightarrow (q_f, 1, >)$
- Estado inicial: q_0
- Estados de aceitação: $\{q_f\}$
- **Testes:**
 - Palavra "111" gera "1000".
 - Palavra "101" gera "110".
 - Palavra "0" gera "1".

O Tipo de Linguagem da Máquina de Turing

A Máquina de Turing é uma das mais poderosas ferramentas da teoria da computação, capaz de simular qualquer algoritmo computável. Ela pode ser classificada como linguagem recursivamente enumerável, pois pode processar todas as cadeias de entrada possíveis e aceitar aquelas que pertencem à linguagem. Se uma cadeia está na linguagem, a máquina eventualmente a aceitará, e se não estiver, a máquina pode continuar rodando indefinidamente.