



UNIVERSIDADE ESTADUAL DO PARANÁ - *CAMPUS* APUCARANA

**Theo Okagawa Rodrigues**

## **RELATÓRIO TÉCNICO - LFA**

**APUCARANA – PR  
2024**



**Theo Okagawa Rodrigues**

## **RELATÓRIO TÉCNICO – LFA**

Trabalho apresentado à disciplina de Linguagens Formais, Autômatos e Computabilidade do curso de Bacharelado em Ciência da Computação.

Professor: Guilherme Henrique de Souza Nakahata;

**APUCARANA – PR  
2024**

## SUMÁRIO

INTRODUÇÃO.....	
CAPÍTULO 1: OBJETIVOS.....	
CAPÍTULO 2: MOTIVAÇÃO E RECURSOS UTILIZADOS.....	
2.1 Motivação.....	
2.1 Estrutura de Dados.....	
2.2 Linguagem de programação e demais informações.....	
CAPÍTULO 3: RESULTADOS.....	
CONCLUSÃO.....	
REFERÊNCIAS.....	

## INTRODUÇÃO

No estudo da ciência da computação, as linguagens formais e autômatos desempenham um papel crucial na compreensão da estrutura e comportamento das linguagens de programação, sistemas computacionais e compiladores. Linguagens formais são conjuntos de cadeias construídas a partir de um alfabeto finito de símbolos e são usadas para definir a sintaxe e semântica de linguagens. Autômatos são modelos matemáticos de máquinas computacionais que operam sobre essas cadeias, realizando operações de reconhecimento e processamento de linguagens.

A Máquina de Turing, introduzida por Alan Turing em 1936, é um modelo computacional mais poderoso, capaz de simular qualquer algoritmo computacional. Diferentemente dos autômatos finitos e de pilha, a Máquina de Turing possui uma fita infinita que serve como uma memória, permitindo a leitura e escrita de símbolos. Este modelo teórico é fundamental para o entendimento do conceito de computabilidade e a delimitação entre problemas que podem ou não ser resolvidos por um algoritmo.

A Máquina de Turing não apenas formaliza a noção de algoritmo, mas também estabelece a base para a teoria da complexidade computacional, que estuda a eficiência dos algoritmos. Este trabalho busca simular uma Máquina de Turing, explorando sua capacidade de processamento e demonstrando sua aplicação na resolução de problemas computacionais.

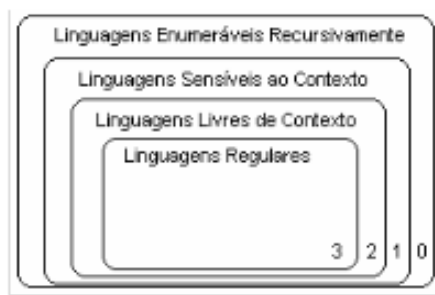


Figura 1

## CAPÍTULO 1

### OBJETIVOS

O objetivo do código apresentado é simular uma Máquina de Turing, um dos conceitos fundamentais em teoria da computação e linguagens formais. Neste trabalho acadêmico, a implementação da Máquina de Turing visa permitir a compreensão prática de como esses conceitos teóricos podem ser aplicados. A simulação facilita a visualização do funcionamento da máquina, desde a leitura de uma palavra de entrada até a manipulação de símbolos na fita e a transição entre estados de acordo com as regras definidas, como exemplificado na figura 2.

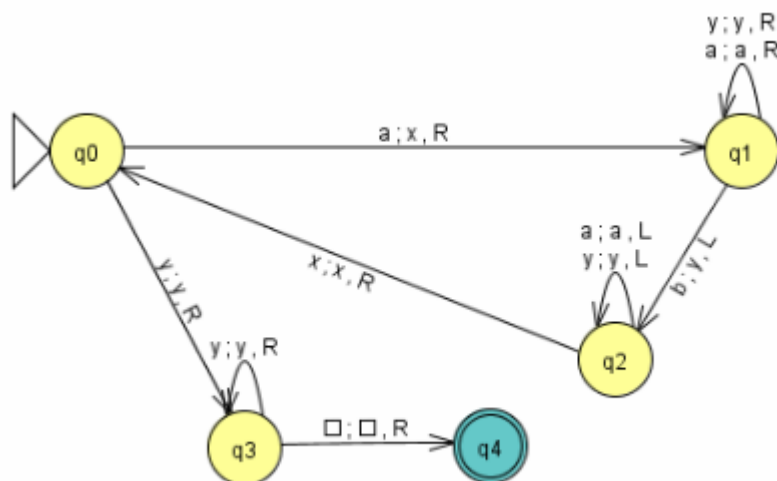


Figura 2

## CAPÍTULO 2

### MOTIVAÇÃO E RECURSOS UTILIZADOS

Baseando-se no que foi anteriormente apresentado, é necessário apresentar os motivos e recursos utilizados para chegar ao resultado esperado.

#### 2.1 Motivação

A implementação de uma simulação de uma Máquina de Turing oferece uma oportunidade única de explorar a interseção entre teoria e prática na ciência da computação. O principal objetivo é tornar o conceito de Máquina de Turing mais acessível e compreensível através da prática direta. Enquanto o modelo de Turing é fundamental na teoria da computação, sua aplicação prática implementada neste trabalho ajuda a revelar a complexidade e praticidade do processamento de informações e da tomada de decisões computacionais.

- $E$  = Conjunto de estados.
- $\Sigma$  = Alfabeto da Fita.
- $i$  = Estado inicial.
- $F$  = Conjunto de estados finais.
- $\gamma$  = Alfabeto auxiliar da Fita.
- $<$  = Marcador de início.
- $\beta$  = Símbolo branco.
- $\delta$  = Função de transição.

*Figura 3*

Para este fim, é necessário receber informações que possibilitem a criação e teste de uma Máquina de Turing, como exemplificado na figura acima. Portanto, é essencial, com base nas informações sobre objetivos e motivações, detalhar os aspectos mais relevantes relacionados à Estrutura de Dados, Linguagem de Programação e outros fatores envolvidos na implementação do código fonte. Esse detalhamento permitirá uma análise mais profunda das funcionalidades do sistema, facilitando a obtenção de uma conclusão geral.

## 2.2 Estrutura de Dados

A estrutura de dados utilizada na simulação da Máquina de Turing é projetada para refletir os componentes essenciais do modelo teórico de Turing e para gerenciar as operações da máquina de forma eficiente. A seguir, descrevemos os principais elementos da estrutura de dados e sua função dentro do código:

### 2.2.1 Variáveis de Configuração:

- `qtd_letras` e `alfabeto`: Armazenam o conjunto de símbolos que a máquina pode processar. `qtd_letras` representa a quantidade de símbolos no alfabeto, enquanto `alfabeto` é uma lista que contém cada símbolo individualmente.
- `qtd_alfabeto_aux` e `alfabeto_aux`: Similarmente, `qtd_alfabeto_aux` representa a quantidade de símbolos no alfabeto auxiliar, e `alfabeto_aux` armazena outros símbolos que serão utilizados para o processamento na fita.
- `qtd_estados` e `estados`: Define quantidade e conjunto de estados da máquina. `qtd_estados` é o número total de estados, e `estados` é uma lista que contém os nomes dos estados.
- `estado_inicial`: Armazena o estado inicial da máquina, onde a execução começa.
- `qtd_estados_finais` e `estados_finais`: Representam os estados finais da máquina, que determinam se a palavra é aceita. `qtd_estados_finais` é o número de estados finais e `estados_finais` é uma lista contendo esses estados.

### 2.2.2 Marcadores:

- `marcador_inicio` e `marcador_branco`: Símbolos especiais utilizados para marcar o início e o espaço em branco da fita. O `marcador_inicio` é colocado no início da fita antes da palavra de entrada, enquanto o `marcador_branco` é utilizado para preencher espaços além da palavra na fita.

### 2.2.3 Função de Transição:

- `funcao_Transicao`: Um dicionário que armazena as regras de

transição da Máquina de Turing. A chave do dicionário é uma tupla composta pelo par (estado, símbolo), e o valor é uma tupla (próximo\_estado, símbolo\_para\_escrever, direcao). Esse dicionário define como a máquina deve se comportar ao ler um símbolo em um estado específico, incluindo o próximo estado, o símbolo a ser escrito na fita e a direção do movimento da cabeça de leitura/escrita.

#### **2.2.4 Fita e Cabeça de Leitura/Escrita:**

- fita: Uma lista que simula a fita infinita da Máquina de Turing. A fita é inicializada com o símbolo de início, seguido pela palavra de entrada e preenchida com símbolos de espaço em branco para permitir que a cabeça se mova além do comprimento da palavra.
- posicao\_cabeca: Um índice que representa a posição atual da cabeça de leitura/escrita na fita. Inicialmente, a cabeça está posicionada logo após o símbolo de início da fita.

#### **2.2.5 Entrada e Verificação:**

- entrada\_usuario: A palavra a ser verificada, convertida em uma lista de caracteres. A verificação garante que todos os caracteres da entrada estejam presentes no alfabeto definido.

Essas estruturas de dados são essenciais para a configuração e a operação da Máquina de Turing, permitindo a modelagem precisa do comportamento da máquina e a execução das regras de transição definidas pelo usuário. A organização e o gerenciamento dessas estruturas garantem a eficácia da simulação e a correta interpretação das palavras de entrada conforme as regras estabelecidas.

### **2.3 Linguagem de Programação e demais informações**

Para a implementação da simulação da Máquina de Turing, a linguagem escolhida foi o Python. Tal linguagem foi selecionada devido a várias características que o tornam particularmente adequado para este tipo de projeto:

- **Sintaxe Clara e Intuitiva:** A sintaxe do Python é conhecida por sua



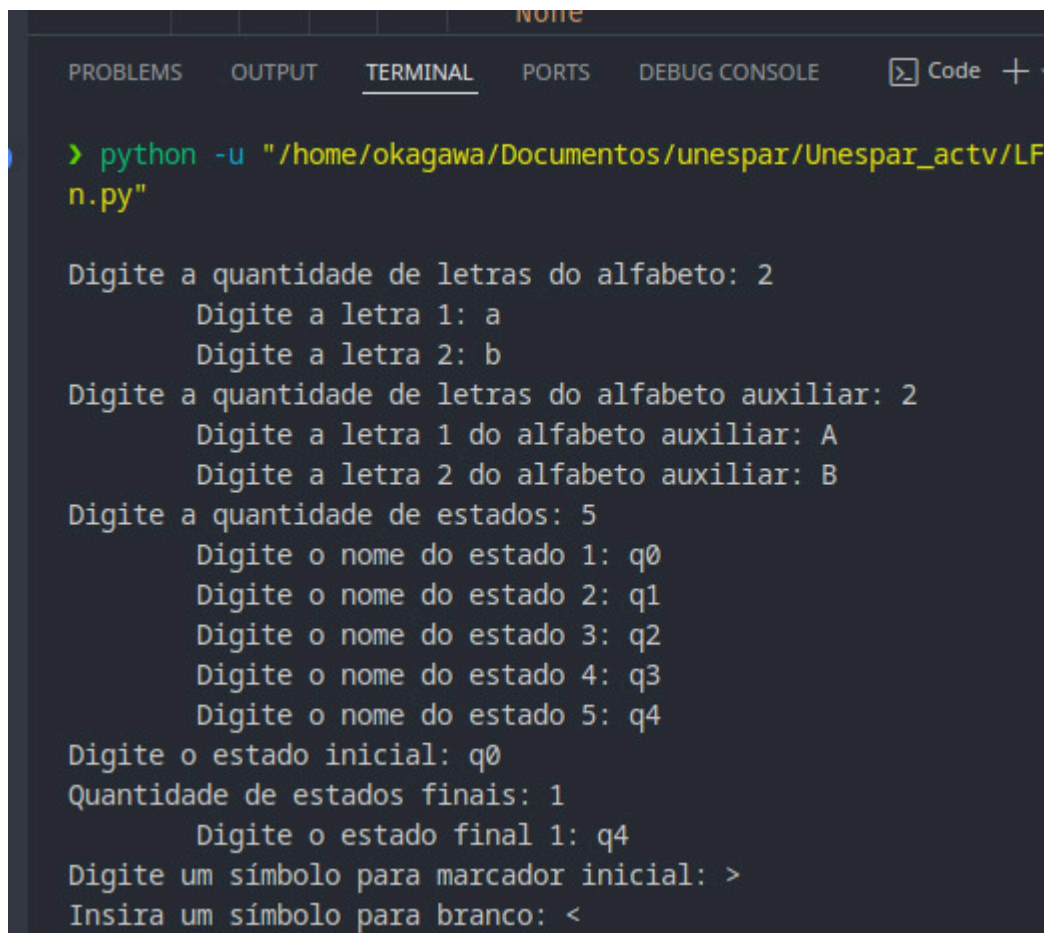
clareza e simplicidade, o que facilita a leitura e a escrita do código. Essa característica é especialmente importante em um projeto acadêmico, onde a compreensão do código é crucial para a análise e a documentação.

- **Facilidade de Manipulação de Dados:** Python oferece suporte robusto para estruturas de dados como listas, dicionários e conjuntos, que são essenciais para a implementação de uma Máquina de Turing. A manipulação eficiente dessas estruturas permite a gestão e a execução de transições e estados de maneira organizada.
- **Bibliotecas e Recursos:** Python possui uma vasta gama de bibliotecas e recursos que podem ser úteis para futuras extensões do projeto. Embora o projeto atual não utilize bibliotecas externas, a capacidade de integrar bibliotecas para análise ou visualização pode ser um benefício adicional.

## CAPÍTULO 3

### RESULTADOS

Com base nos objetivos estabelecidos, o resultado esperado era a criação de um código que demonstrasse como uma Máquina de Turing pode reconhecer ou rejeitar palavras pertencentes a uma linguagem, conforme a sua descrição formal e a tabela de transições. Após a implementação completa e revisão do código, esses objetivos foram plenamente alcançados. O sistema desenvolvido é interativo e funcional, permitindo ao usuário inserir a quantidade de letras do alfabeto, os próprios símbolos do alfabeto, a representação dos estados, o número total de estados, a quantidade de estados finais, o estado inicial, a lista de estados finais, o marcador de início, o símbolo de espaço em branco e as transições, que incluem o estado futuro, o símbolo a ser escrito e a direção (caso não seja anulada). O funcionamento do exemplo de Máquina de Turing é detalhado nas Figuras 4, 5, 6, 7.



```

> python -u "/home/okagawa/Documentos/unepar/Unepar_actv/LFN.py"

Digite a quantidade de letras do alfabeto: 2
    Digite a letra 1: a
    Digite a letra 2: b
Digite a quantidade de letras do alfabeto auxiliar: 2
    Digite a letra 1 do alfabeto auxiliar: A
    Digite a letra 2 do alfabeto auxiliar: B
Digite a quantidade de estados: 5
    Digite o nome do estado 1: q0
    Digite o nome do estado 2: q1
    Digite o nome do estado 3: q2
    Digite o nome do estado 4: q3
    Digite o nome do estado 5: q4
Digite o estado inicial: q0
Quantidade de estados finais: 1
    Digite o estado final 1: q4
Digite um símbolo para marcador inicial: >
Insira um símbolo para branco: <
  
```

Figura 4

Preencha as transições:

```

 $\delta(q_0, a): q_1, A, R$ 
 $\delta(q_0, b): 2, B, R$ 
 $\delta(q_0, A): q_1, A, R$ 
 $\delta(q_0, B): q_2, B, R$ 
 $\delta(q_0, >): q_0, >, R$ 
 $\delta(q_0, <): q_0, <, R$ 
 $\delta(q_1, a): q_1, A, R$ 
 $\delta(q_1, b): q_2, B, R$ 
 $\delta(q_1, A): q_1, A, R$ 
 $\delta(q_1, B): q_2, B, R$ 
 $\delta(q_1, >): X$ 
 $\delta(q_1, <): q_3, <, L$ 
 $\delta(q_2, a): q_1, A, R$ 
 $\delta(q_2, b): q_2, B, R$ 
 $\delta(q_2, A): q_1, A, R$ 
 $\delta(q_2, B): q_2, B, R$ 
 $\delta(q_2, >): X$ 
 $\delta(q_2, <): q_4, <, L$ 
 $\delta(q_3, a): q_3, A, L$ 
 $\delta(q_3, b): q_3, B, L$ 
 $\delta(q_3, A): q_3, A, L$ 
 $\delta(q_3, B): q_3, B, L$ 

```

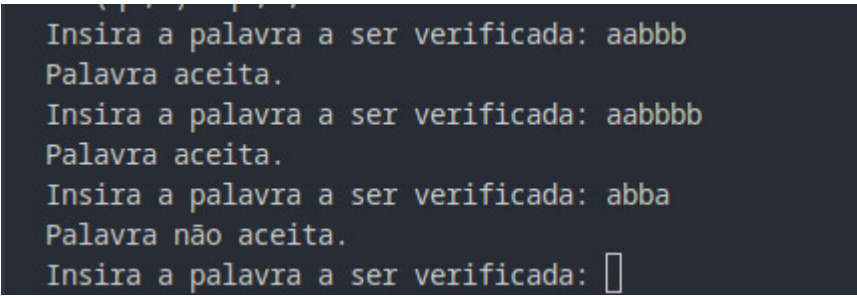
Figura 5

```

 $\delta(q_3, B): q_3, B, L$ 
 $\delta(q_3, >): X$ 
 $\delta(q_3, <): q_4, <, L$ 
 $\delta(q_4, a): q_4, A, L$ 
 $\delta(q_4, b): q_4, B, L$ 
 $\delta(q_4, A): q_4, A, L$ 
 $\delta(q_4, B): q_4, B, L$ 
 $\delta(q_4, >): X$ 
 $\delta(q_4, <): q_4, <, L$ 

```

Figura 6



A terminal window with a dark background and light-colored text. It shows a program that prompts the user to enter a word for verification. The program accepts 'aabb' and 'aabbbb' as valid words, but rejects 'abba'. The prompt is currently waiting for input.

```
(17) (17)  
Insira a palavra a ser verificada: aabb  
Palavra aceita.  
Insira a palavra a ser verificada: aabbbb  
Palavra aceita.  
Insira a palavra a ser verificada: abba  
Palavra não aceita.  
Insira a palavra a ser verificada: 
```

Figura 7

## CONCLUSÃO

Com base no que foi discutido, é evidente que a implementação prática, mesmo em disciplinas mais teóricas, pode ser altamente vantajosa e apropriada. Ela não só proporciona uma compreensão mais profunda de conceitos externos à disciplina, mas também contribui significativamente para a consolidação do conhecimento adquirido em sala de aula.

O projeto alcançou com sucesso os objetivos de demonstrar o funcionamento de uma Máquina de Turing e validar seu comportamento através de uma simulação prática. A experiência adquirida na implementação e na análise da máquina oferece uma base sólida para futuras investigações e desenvolvimentos na área da ciência da computação. A compreensão aprofundada do modelo de Turing e a aplicação prática obtida através deste trabalho são contribuições significativas para o estudo dos fundamentos teóricos e das técnicas computacionais avançadas.