



UNIVERSIDADE ESTADUAL DO PARANÁ - CAMPUS APUCARANA

NOME DO(S) ALUNO(S)
Gabriel Ricetto Da Rocha

MAQUINA DE TURING

APUCARANA – PR
2024

Gabriel Ricetto Da Rocha

MAQUINA DE TURING

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

Professor: Guilherme Nakahata

APUCARANA – PR 2024

SUMÁRIO

INTRODUÇÃO	3
CAPÍTULO 1: OBJETIVO	4
CAPÍTULO 2: MOTIVAÇÕES E RECURSOS UTILIZADOS	6
2.1 Estrutura de dados	7
2.2 Linguagens da programação	7
2.3 Bibliotecas	7
CAPÍTULO 3: RESULTADOS	6
3.1 Testes na prática	6
CONCLUSÃO	8
REFERÊNCIAS	9

INTRODUÇÃO

As Máquinas de Turing são fundamentais na teoria da computação e na engenharia, ajudando-nos a entender e modelar sistemas computacionais que operam de maneira sequencial e discreta. Elas são especialmente úteis para tarefas complexas como a análise sintática e a verificação formal de sistemas. Neste relatório, exploraremos a implementação de uma Máquina de Turing, discutindo seus componentes, funcionamento e importância. Nosso objetivo é fornecer uma compreensão acessível e prática sobre essa poderosa ferramenta teórica.

CAPÍTULO 1

OBJETIVO

O objetivo do estudo das Máquinas de Turing é proporcionar uma compreensão sólida e aplicável dessa ferramenta essencial na teoria da computação. As Máquinas de Turing são utilizadas para modelar algoritmos e problemas computacionais complexos, servindo como uma base teórica para a definição de computabilidade e complexidade. Especificamente, o estudo das Máquinas de Turing busca:

1. **Modelagem de Sistemas:** Facilitar a representação de sistemas computacionais que podem ser descritos por um conjunto de estados e transições.
2. **Validação e Reconhecimento:** Permitir a verificação de cadeias de entrada contra padrões predefinidos, garantindo que apenas as sequências válidas sejam aceitas.
3. **Simplificação de Problemas:** Auxiliar na decomposição e solução de problemas complexos através da estruturação em estados e transições.
4. **Educação e Formação:** Servir como uma ferramenta pedagógica para ensinar conceitos fundamentais da teoria da computação.
5. **Desenvolvimento de Software:** Contribuir para o desenvolvimento de software robusto e eficiente, fornecendo uma base teórica para algoritmos complexos.

CAPÍTULO 2

MOTIVAÇÕES E RECURSOS UTILIZADOS

3.1 Estrutura de dados

Para a implementação de uma Máquina de Turing, utilizamos as seguintes estruturas de dados:

- Matrizes de Transição: Para armazenar as transições de estado.
- Vetores: Para armazenar os estados finais e os símbolos do alfabeto.
- Variáveis Inteiras: Para gerenciar os estados atuais e índices de símbolos.

3.2 Linguagens de Programação

Java: Escolhida por sua robustez e capacidade de gerenciamento de memória automática, além de ser uma linguagem amplamente utilizada e de fácil compreensão

3.3 Bibliotecas

Utilizamos as bibliotecas padrão do Java para manipulação de entrada e saída, além de manipulação de cadeias de caracteres.

CAPÍTULO 3

RESULTADOS

Mediante os objetivos apresentados, implementamos um código Java que exemplifica como uma Máquina de Turing processa uma palavra de entrada conforme uma descrição formal e tabela de transições.

```
Digite a quantidade de letras do alfabeto: 2
Digite a 1a letra: a
Digite a 2a letra: b

Digite a quantidade de letras do alfabeto familiar: 2
Digite a 1a letra: A
Digite a 2a letra: B

Digite a quantidade de estados: 4

Digite o estado inicial: 1

Digite a quantidade de estados finais: 1

Digite o 1o estado final: 4

Digite um marcador de inicio: >

Defina um simbolo para vazio: <
Digite a palavra a ser testada: aabbb
```

```
obs: Digite -1 para quando nao houver transicao
Digite as transicoes:
Digite o estado futuro da transicao: 1,1
2
Digite o alfabeto futuro da transicao: 1,1
A
Digite a direcao da transicao: 1,1 (D para Direita ou E para Esquerda)
D
-----
Digite o estado futuro da transicao: 1,2
-1
O campo sera anulado!
-----
Digite o estado futuro da transicao: 1,3
-1
O campo sera anulado!
-----
Digite o estado futuro da transicao: 1,4
-1
O campo sera anulado!
```

```

-----
Digite o estado futuro da transicao: 2,1
2
Digite o alfabeto futuro da transicao: 2,1
A
Digite a direcao da transicao: 2,1 (D para Direita ou E para Esquerda)
D
-----
Digite o estado futuro da transicao: 2,2
3
Digite o alfabeto futuro da transicao: 2,2
B
Digite a direcao da transicao: 2,2 (D para Direita ou E para Esquerda)
D
-----
Digite o estado futuro da transicao: 2,3
-1
O campo sera anulado!
-----
Digite o estado futuro da transicao: 2,4
-1
O campo sera anulado!
-----
Digite o estado futuro da transicao: 3,1
-1
O campo sera anulado!
-----
Digite o estado futuro da transicao: 3,2
3
Digite o alfabeto futuro da transicao: 3,2
B
Digite a direcao da transicao: 3,2 (D para Direita ou E para Esquerda)
D
-----
Digite o estado futuro da transicao: 3,3
-1
O campo sera anulado!
-----
Digite o estado futuro da transicao: 3,4
-1
O campo sera anulado!
-----
Digite o estado futuro da transicao: 4,1

```


CONCLUSÃO

Com o desenvolvimento e aprimoramento do código da Máquina de Turing, conseguimos criar uma ferramenta eficaz para validar cadeias de caracteres conforme uma linguagem formal específica. A implementação foi pensada para ser intuitiva e funcional, facilitando o entendimento dos conceitos fundamentais de teoria da computação.

REFERÊNCIAS

Wikipedia: https://pt.wikipedia.org/wiki/M%C3%A1quina_de_Turing

Github: <https://github.com/GuilhermeNakahata/UNESPAR-2024/tree/main/Linguagens%20Formais%20Automatos%20e%20Computabilidade>