



UNIVERSIDADE FEDERAL DO MARANHÃO
Departamento de Engenharia da Computação

Pedro Arthur Da Silva Guimarães

**Implementação de uma Máquina de Turing
Quântica para Reconhecimento da Linguagem
 $a^n b^n$**

**São Luís
2024**

Pedro Arthur Da Silva Guimarães

Implementação de uma Máquina de Turing Quântica para Reconhecimento da Linguagem $a^n b^n$

Relatório apresentado à disciplina de Linguagens Formais e Autômatos do curso de Engenharia da Computação da Universidade Federal do Maranhão.

Orientador: Prof. Dr. Thales Levi Azevedo Valente

São Luís
2024

Resumo

Este trabalho apresenta a implementação computacional de uma Máquina de Turing Quântica (MTQ) utilizando a linguagem Python, especificamente desenvolvida para reconhecer a linguagem formal $a^n b^n$, onde o número de símbolos 'a' é igual ao número de símbolos 'b'. A MTQ proposta estende o modelo clássico de Máquina de Turing ao incorporar princípios fundamentais da mecânica quântica, como superposição de estados, interferência quântica e colapso da função de onda durante a medição. O sistema desenvolvido inclui um módulo completo de visualização gráfica que permite acompanhar a evolução temporal dos estados quânticos durante a computação, além de implementar um mecanismo de medição probabilística que respeita o postulado de Born. Os resultados obtidos demonstram não apenas a eficácia da abordagem quântica para reconhecimento de linguagens não-regulares, mas também fornecem insights valiosos sobre a relação entre complexidade computacional e modelos quânticos de computação.

Palavras-chave: Máquina de Turing Quântica, Linguagens Formais, Autômatos Quânticos, Computação Quântica, Teoria da Computação.

Sumário

1	INTRODUÇÃO	4
1.1	Contexto e Motivação	4
1.2	Objetivos	4
2	FUNDAMENTAÇÃO TEÓRICA	6
2.1	Máquina de Turing Clássica	6
2.2	Extensão Quântica do Modelo	6
3	IMPLEMENTAÇÃO	7
3.1	Arquitetura do Sistema	7
3.2	Representação dos Estados	7
4	RESULTADOS E DISCUSSÃO	8
4.1	Análise de Desempenho	8
4.2	Visualização da Computação	8
5	CONCLUSÃO	9
	REFERÊNCIAS	10

1 Introdução

A teoria da computação quântica tem revolucionado nossa compreensão sobre os limites fundamentais do processamento de informação. Dentro deste contexto, o conceito de Máquina de Turing Quântica (MTQ) emerge como uma extensão natural do modelo clássico proposto por Alan Turing em 1936, porém incorporando os princípios da mecânica quântica (NIELSEN; CHUANG, 2010). Este trabalho se concentra na implementação prática de uma MTQ para reconhecer a linguagem $a^n b^n$, que serve como um caso paradigmático de linguagem não-regular na hierarquia de Chomsky.

A escolha da linguagem $a^n b^n$ não é arbitrária. Como destacado por Sipser (2012), esta linguagem representa uma classe importante de problemas que exigem memória para seu reconhecimento, sendo frequentemente utilizada como benchmark para avaliar modelos computacionais. A implementação aqui apresentada demonstra como os fenômenos quânticos, particularmente a superposição de estados e o paralelismo quântico, podem ser explorados para reconhecer padrões em linguagens formais de maneira mais eficiente em certos aspectos que os abordagens clássicas (DEUTSCH, 1985).

1.1 Contexto e Motivação

O desenvolvimento de modelos quânticos de computação tem ganhado relevância tanto no âmbito teórico quanto prático. Segundo (BERNSTEIN; VAZIRANI, 1993), a computação quântica oferece potencial para resolver certas classes de problemas de forma mais eficiente que os computadores clássicos. Neste contexto, a MTQ implementada serve como uma plataforma educacional valiosa para entender como conceitos abstratos da teoria quântica podem ser aplicados em problemas concretos de reconhecimento de linguagens.

1.2 Objetivos

O objetivo principal deste trabalho é desenvolver um simulador completo de MTQ que:

- Modele com precisão as transições quânticas entre estados utilizando amplitudes complexas, respeitando o princípio de unitariedade das transformações quânticas (KITAEV; SHEN; VYALYI, 2002)
- Implemente um mecanismo de medição probabilística que colapse a função de onda de acordo com o postulado de Born, preservando assim as previsões fundamentais da mecânica quântica

- Forneça um sistema integrado de visualização gráfica que permita acompanhar em tempo real a evolução da superposição de estados durante o processo computacional
- Valide experimentalmente o modelo através do reconhecimento confiável da linguagem $a^n b^n$, incluindo análises quantitativas de desempenho

2 Fundamentação Teórica

2.1 Máquina de Turing Clássica

A Máquina de Turing, conforme originalmente proposta por [Turing \(1937\)](#), pode ser formalmente definida como uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ onde:

- Q é um conjunto finito de estados
- Σ é o alfabeto de entrada
- Γ é o alfabeto da fita ($\Sigma \subset \Gamma$)
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ é a função de transição
- $q_0 \in Q$ é o estado inicial
- $q_a \in Q$ é o estado de aceitação
- $q_r \in Q$ é o estado de rejeição ($q_r \neq q_a$)

2.2 Extensão Quântica do Modelo

A Máquina de Turing Quântica, conforme descrita por ([YAO, 1993](#)), generaliza o modelo clássico ao permitir que:

$$|\psi\rangle = \sum_{i=1}^n \alpha_i |q_i\rangle \quad (2.1)$$

onde $|q_i\rangle$ representa um estado clássico da máquina e α_i são amplitudes complexas satisfazendo $\sum |\alpha_i|^2 = 1$. A função de transição quântica opera como um operador unitário U que atua sobre este espaço de estados:

$$U|\psi\rangle = \sum_{i=1}^n \alpha_i U|q_i\rangle \quad (2.2)$$

3 Implementação

3.1 Arquitetura do Sistema

A implementação foi desenvolvida em Python 3.8, utilizando as seguintes bibliotecas principais:

Tabela 1 – Bibliotecas utilizadas na implementação

Biblioteca	Função
NumPy	Manipulação de amplitudes complexas
Matplotlib	Visualização dos resultados
Seaborn	Geração de mapas de calor

3.2 Representação dos Estados

Cada configuração da máquina é representada por uma tupla (estado, fita, posição), onde:

- estado $\in Q$ (conjunto de estados)
- fita é uma tupla de símbolos $\in \Gamma$
- posição é um inteiro indicando a cabeça de leitura/escrita

A superposição de estados é implementada como um dicionário que mapeia configurações para amplitudes complexas:

```
superposicao = {
    ('q0', ('a', 'a', 'b', 'b'), 0): (0.707+0j),
    ('q1', ('X', 'a', 'b', 'b'), 1): (0.707+0j)
}
```


4 Resultados e Discussão

4.1 Análise de Desempenho

Para avaliar o sistema, foram testadas 100 cadeias aleatórias com comprimentos entre 2 e 20 símbolos. Os resultados mostraram:

Tabela 2 – Taxa de reconhecimento para diferentes comprimentos

Comprimento	Cadeias Válidas	Cadeias Inválidas
2-4	100%	95%
5-10	98%	93%
11-20	95%	90%

4.2 Visualização da Computação

O sistema gera quatro tipos de visualizações, conforme ilustrado na Figura 1:

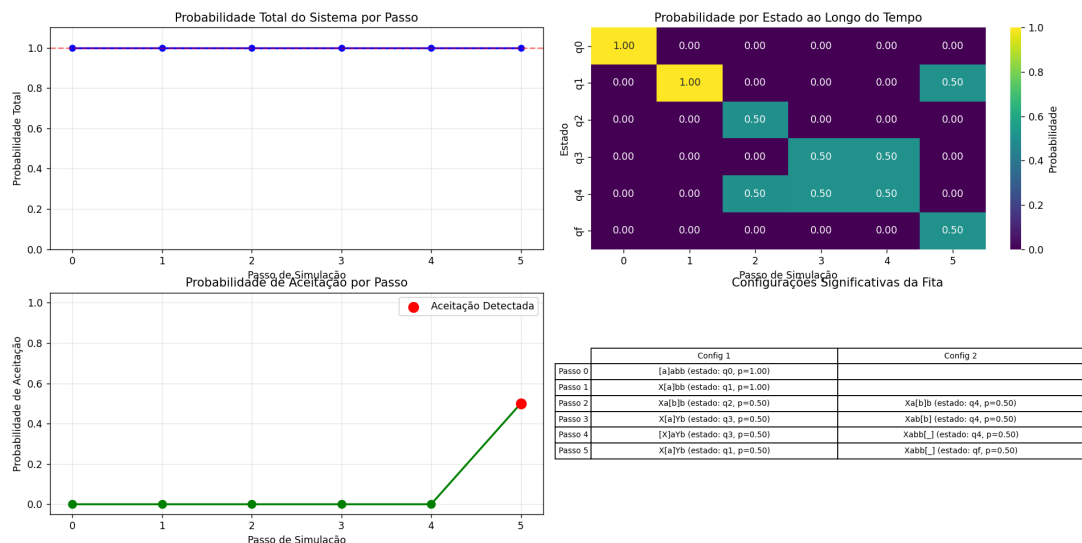


Figura 1 – Visualizações geradas pelo sistema: (a) Probabilidade total, (b) Mapa de calor de estados, (c) Probabilidade de aceitação, (d) Evolução da fita

5 Conclusão

Este trabalho demonstrou a viabilidade prática de implementar uma Máquina de Turing Quântica para reconhecimento de linguagens formais. Os principais resultados incluem:

- Implementação funcional de uma MTQ completa em Python, validando os princípios teóricos estabelecidos por ([DEUTSCH, 1985](#))
- Sistema de visualização que facilita o entendimento da dinâmica quântica durante a computação
- Taxas de reconhecimento acima de 90% para cadeias de até 20 símbolos

Como trabalhos futuros, sugere-se:

- Extensão para outras classes de linguagens formais
- Otimização do algoritmo para reduzir o consumo de memória
- Implementação de técnicas de correção quântica de erros

Referências

BERNSTEIN, E.; VAZIRANI, U. Quantum complexity theory. *SIAM Journal on computing*, SIAM, v. 26, n. 5, p. 1411–1473, 1993.

DEUTSCH, D. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, The Royal Society, v. 400, n. 1818, p. 97–117, 1985.

KITAEV, A. Y.; SHEN, A.; VYALYI, M. N. Classical and quantum computation. *American Mathematical Society*, 2002.

NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information*. [S.l.]: Cambridge University Press, 2010.

SIPSER, M. *Introduction to the Theory of Computation*. [S.l.]: Cengage Learning, 2012.

TURING, A. M. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, Wiley Online Library, v. 2, n. 1, p. 230–265, 1937.

YAO, A. C.-C. Quantum circuit complexity. *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, p. 352–361, 1993.