

Linguagens Formais e Autômatos

Engenharia da
Computação

Alunos: Victor Ferreira Braga e
William Valther Silva Martins

Profº: Dr Thales Valente

Universidade Federal do Maranhão



Sumário

Introdução
Objetivos
Metodologia
Resultados
Conclusão
Referências

Pág 01

Pág 12

Pág 13

Pág 16

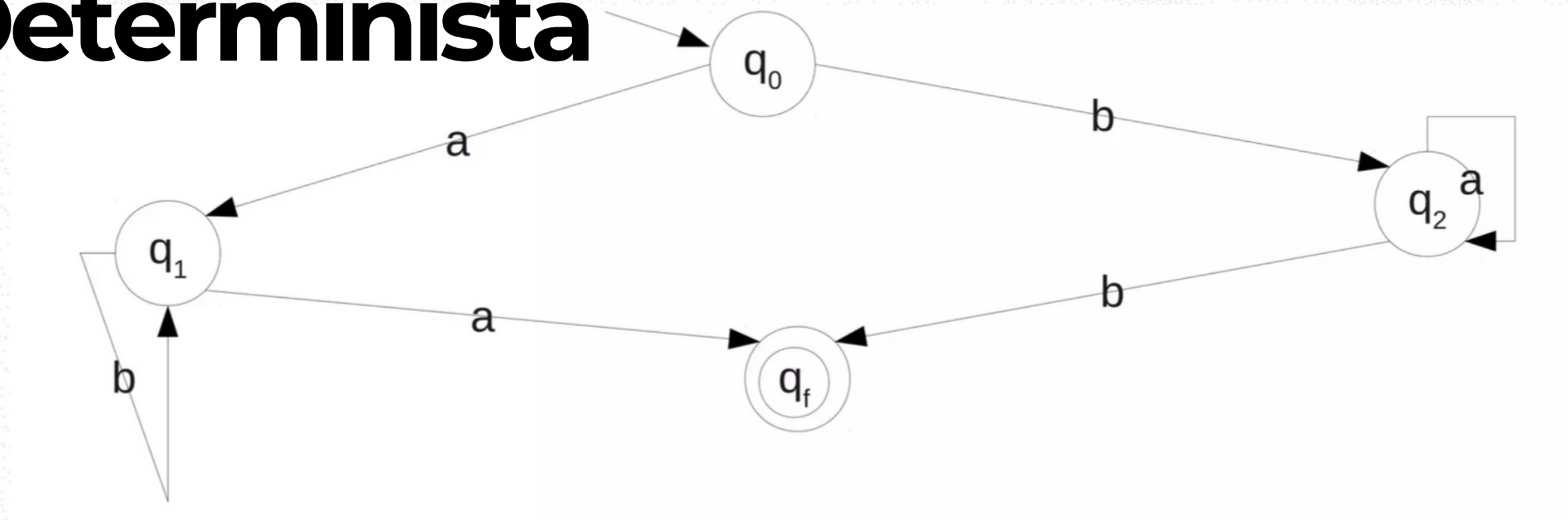
Pág 18

Pág 19

Introdução à Máquina de Turing Não Determinista

MT

Determinista

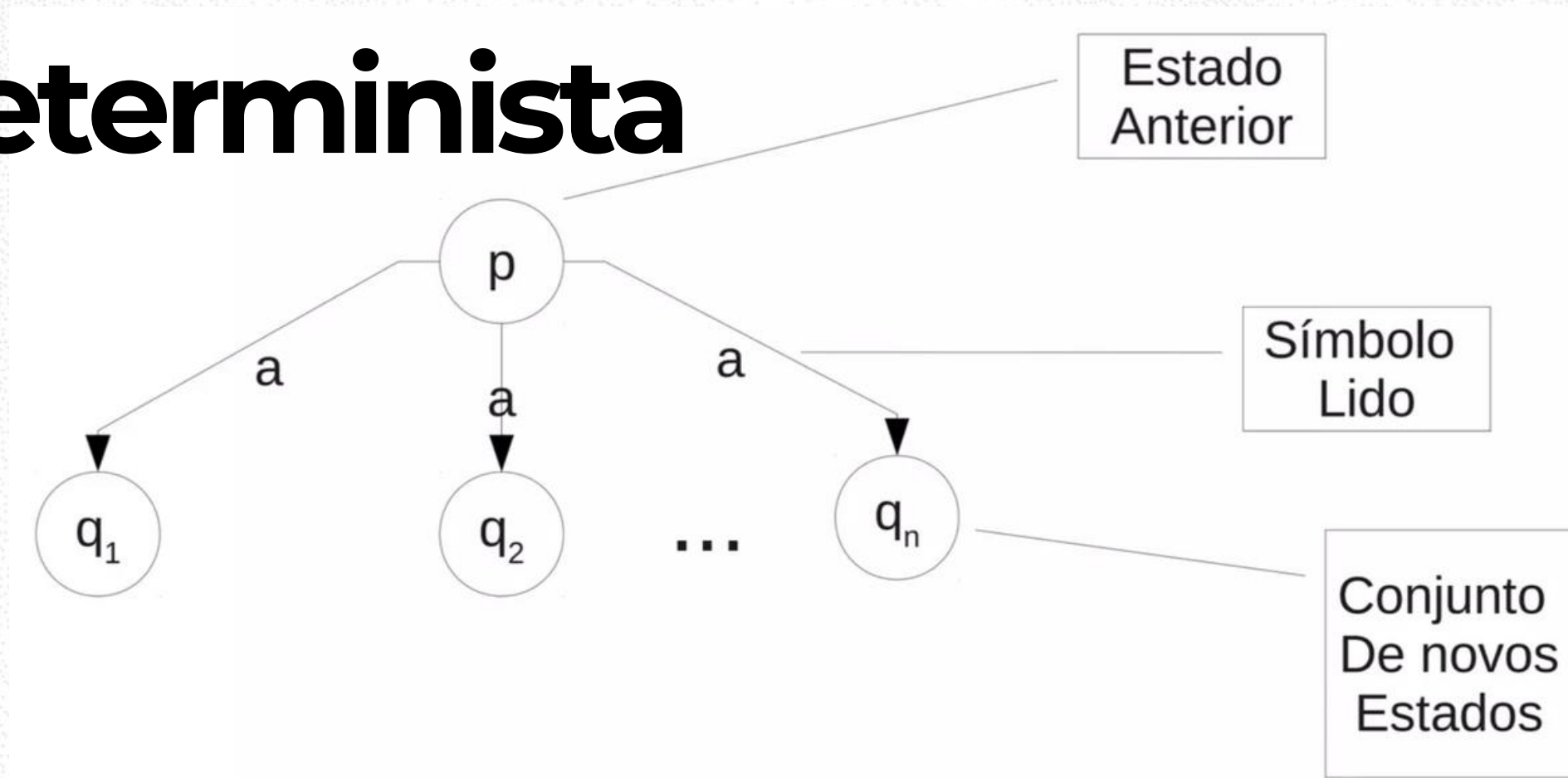


Para cada configuração (estado atual + símbolo lido), há exatamente uma transição possível

Exemplo

Se a MT está no estado q_0 e lê o símbolo 1, ela sempre executará a mesma transição.

MT Não Determinista



A máquina pode escolher entre múltiplas transições possíveis para um mesmo estado e símbolo

Isso significa que ela pode “explorar diferentes caminhos” ao mesmo tempo

Repare que quando estamos no estado p e lemos o símbolo a podemos ir para q_1, q_2 ou q_n . Essa é a razão do não determinismo.

Simulação da MT

Fita limitada

- Em uma MT tradicional, a fita é infinita
- Aqui, a fita é restringida ao tamanho da entrada
- Por quê? Isso impede a máquina de explorar espaços ilimitados, tornando a simulação mais eficiente

Arestas ponderadas

- Cada transição entre estados tem um peso associado
- Esse peso pode representar:
 - Tempo de execução
 - Custo computacional
 - Número de operações realizadas

Simulação da MT

Troca de BFS por Dijkstra

- BFS encontra o caminho mais curto em termos de número de passos, mas ignora os pesos.
- Dijkstra encontra o caminho de menor custo, levando em conta os pesos das transições.

Representação da Máquina Como Grafo Ponderado

Definição formal da Máquina de Turing

Uma MT pode ser representada como uma 8-tupla:

$$M = (\Sigma, Q, \delta, q_0, F, V, \odot, \dagger)$$

Onde:

Σ : Alfabeto da fita.

Q : Conjunto de estados

δ : Função de transição (regras de mudança de estado)

q_0 : Estado inicial

F : Conjunto de estados de aceitação

V : Símbolos da fita

\odot e \dagger : Delimitadores de início e fim da fita

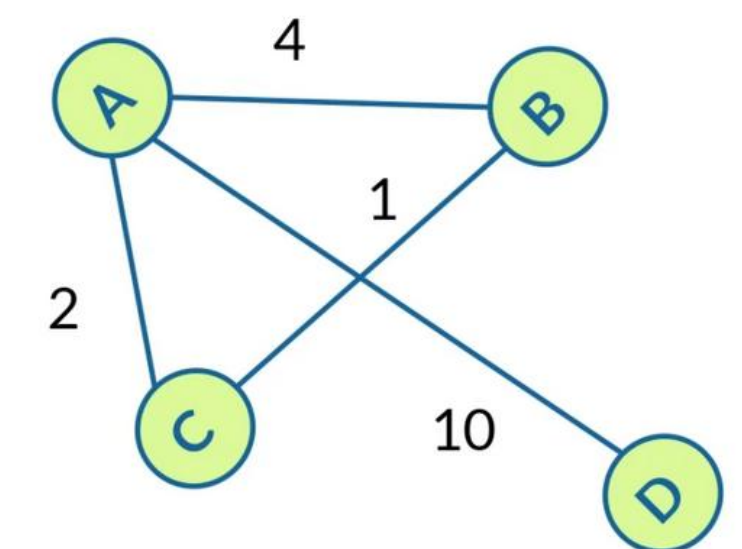
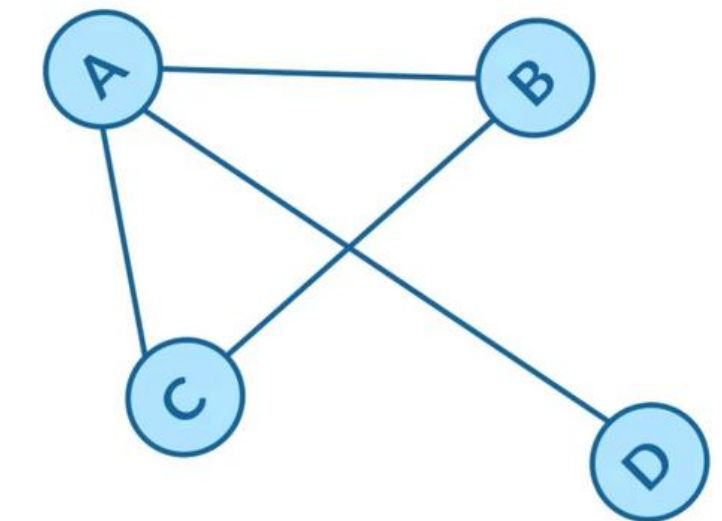
Essa estrutura define como a máquina funciona e como se move entre os estados

Transformação da MT em um Grafo

Estados como Vértices: Cada estado da máquina é representado como um nó no grafo.

Transições como Arestas: As transições entre estados são representadas por arestas dirigidas.

Pesos nas Arestas: Os pesos correspondem ao custo associado a cada transição.

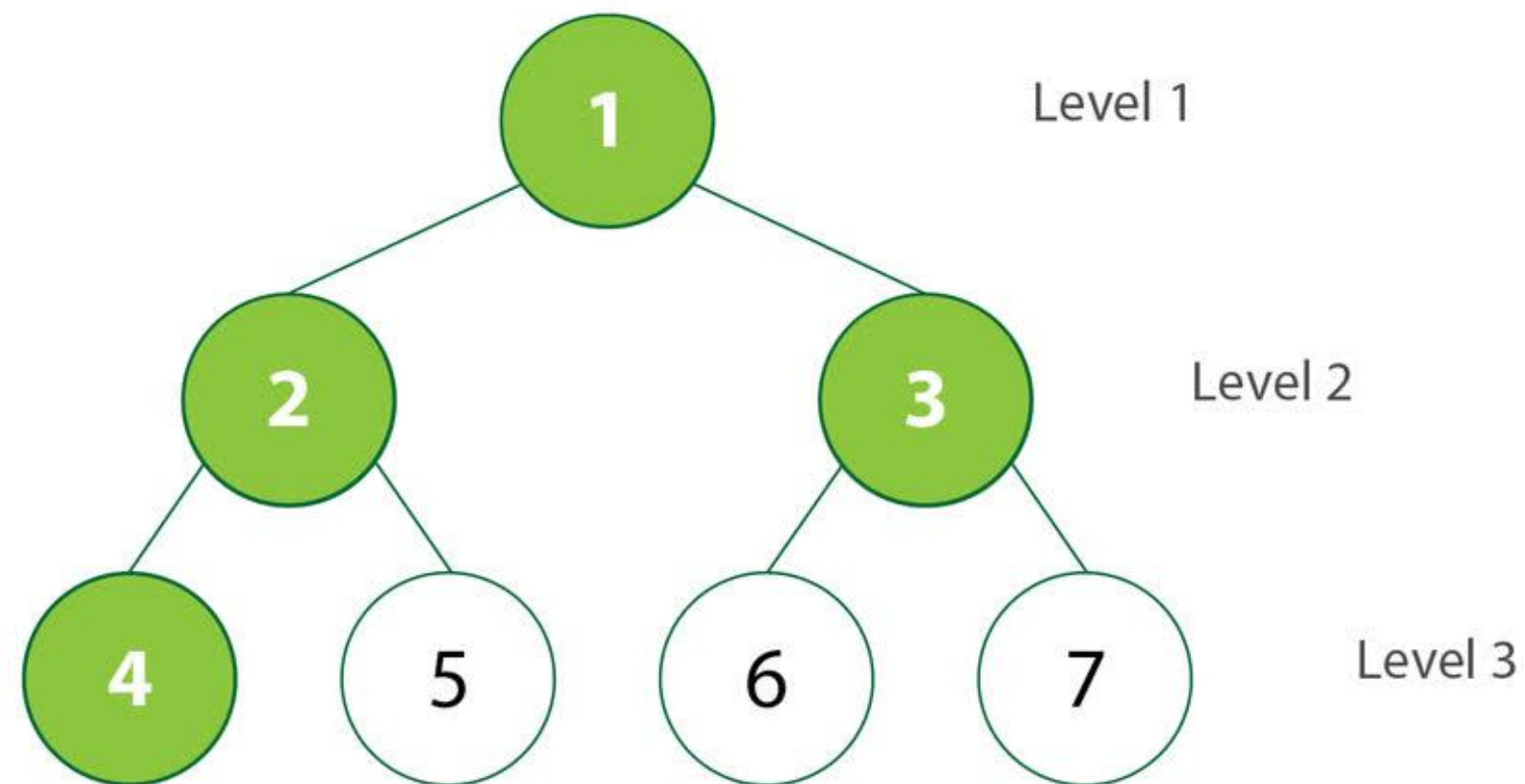


Problema do BFS e a Alternativa com Dijkstra

Limitações do BFS

Busca em Largura (BFS): Explora todos os nós em um nível antes de avançar para o próximo.

Limitação: O BFS não leva em conta os pesos das arestas. Como consequência, ele pode escolher um caminho que tem mais passos, mas não necessariamente o de menor custo.



Vantagens do Algoritmo de Dijkstra

Dijkstra é mais eficiente porque:

1. Leva em conta os pesos das transições
2. Garante o caminho de menor custo até um estado final
3. Evita caminhos desnecessários, sendo mais otimizado

Como funciona?

- Criamos uma fila de prioridade
- Sempre exploramos o menor custo acumulado
- O primeiro estado de aceitação encontrado é garantidamente o de menor custo

Objetivos

Objetivos a serem alcançados:

- Implementar uma Máquina de Turing Não Determinista com uma fita limitada.
- Substituir a busca em largura (BFS) por um algoritmo de menor custo (Dijkstra) para encontrar o caminho com o menor peso até um estado de aceitação.
- Fornecer como resultado:
 - Fita final após a execução.
 - Caminho percorrido pela máquina.
 - Soma ponderada das transições.

Metodologia

Teoria da Computação

- A Máquina de Turing é modelada como um autômato finito não determinístico com uma fita e transições.

Definição formal dos componentes:

- Estados ($q_0, q_1, q_2, q_3, q_4, q_{\text{accept}}$).
- Símbolos na fita (0, 1, $_$).
- Cabeçote de leitura/escrita que move para a esquerda (L) ou direita (R).
- Função de transição que determina o próximo estado com base no símbolo lido.

Metodologia

Algoritmo de Grafos

- O conjunto de estados e transições foi modelado como um **grafo direcionado**.
- As **transições possuem pesos**, tornando o problema equivalente a **encontrar o caminho de menor custo**.
- O **algoritmo de Dijkstra** foi utilizado para:
 - Explorar as transições disponíveis a partir de um estado atual.
 - Priorizar caminhos de menor custo usando uma fila de prioridade (heap).
 - Garantir que a busca termine ao alcançar um estado de aceitação.

Metodologia

Visualização e Animação

- A estrutura da Máquina de Turing foi representada com **networkx** para gerar **grafos interativos**.
- A execução foi animada passo a passo utilizando **matplotlib.animation**.

Resultados

Estado Inicial

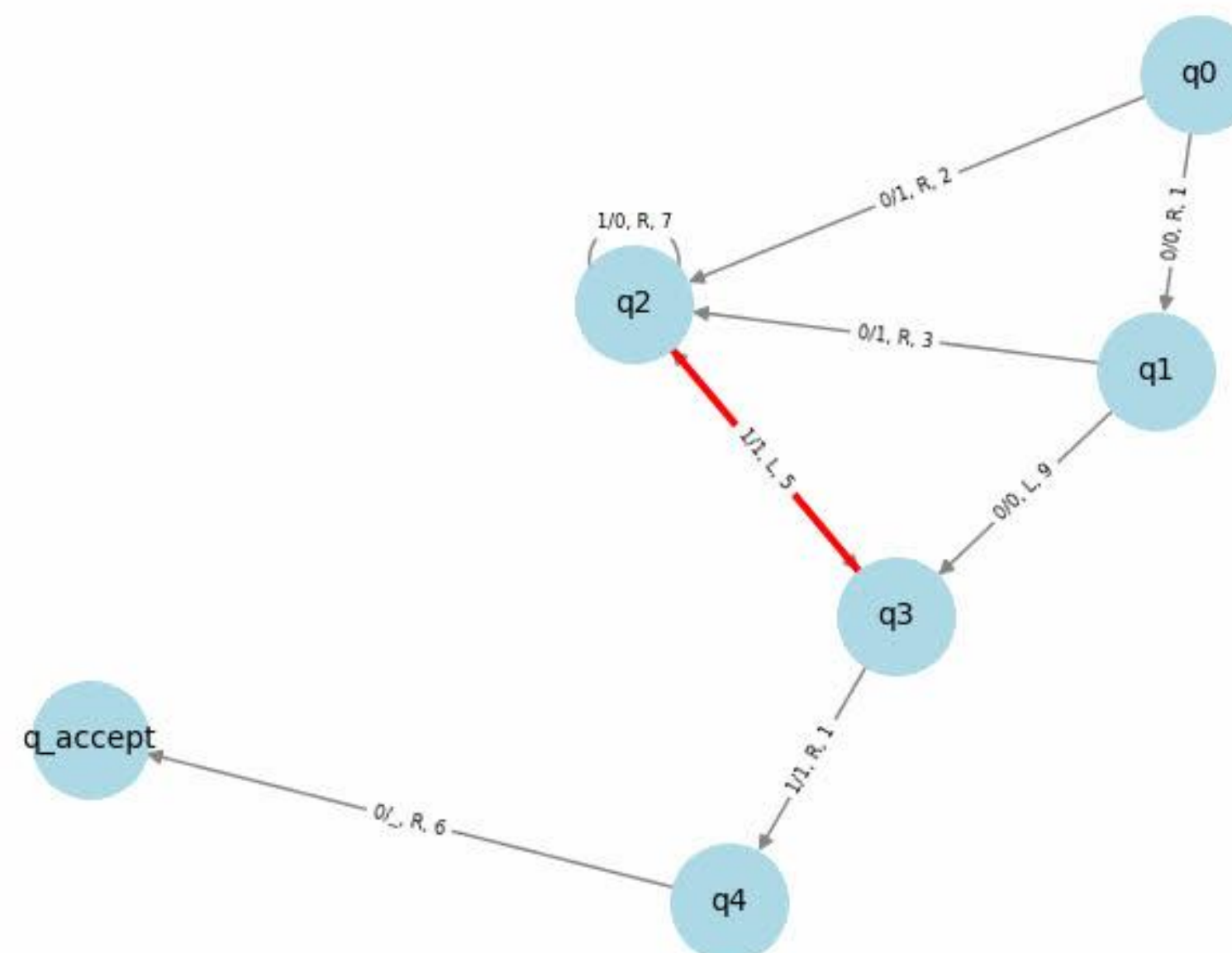
Estados					
q0	q1	q2	q3	q4	q_accept
Fita					
0	0	1	1	0	

Conjunto de transações					
Estado_Atual	Símbolo_Lido	Próximo_Estado	Símbolo_Escrito	Direção	Peso
q0	0	q1	0	R	1
q0	0	q2	1	R	2
q1	0	q2	1	R	3
q1	0	q3	0	L	9
q2	1	q3	0	R	6
q2	1	q2	0	R	7
q3	1	q4	1	R	1
q3	1	q2	1	L	5
q4	0	q_accept	_	R	6

Resultados

Passos				
Nº	Estado	Cabeça	Custo	Fita
1	q0	0	0	00110_
2	q1	1	1	00110_
3	q2	2	4	01110_
4	q3	3	10	01010_
5	q4	4	11	01010_
6	q_accept	5	17	0101_

Passo 4: Estado = q3, Cabeça = 3, Custo = 10
Fita: 01010_



Conclusão

O projeto utilizou uma abordagem **computacional teórica** combinada com **otimização de caminhos em grafos**, garantindo um **modelo eficiente e visualmente interativo** da Máquina de Turing.

O uso de **Dijkstra** garantiu que a **execução fosse otimizada**, priorizando **caminhos de menor custo**. A integração de **visualização dinâmica** tornou o **processo mais intuitivo para análise e depuração**.

Referências

19

1. ELFRMKR98. Getting started with trees: Breadth-First Search (BFS). Medium, 2023. Disponível em: <https://medium.com/@elfrmkr98/getting-started-with-trees-breadth-first-search-bfs-344dc92cbd30>. Acesso em: 11 fev. 2025.
2. RIBEIRO, Leila. Da Máquina de Turing ao Pensamento Computacional. UFRGS, 2012. Disponível em: <https://www.ufrgs.br/alanturingbrasil2012/presentation-LeilaRibeiro-ptBR.pdf>. Acesso em: 11 fev. 2025.
3. VIEIRA, Nelson. Máquinas de Turing. UFMG, 2017. Disponível em: <https://homepages.dcc.ufmg.br/~nvieira/cursos/tl/a17s2/livro/cap4.pdf>. Acesso em: 11 fev. 2025.