

Lista de Exercícios

Máquinas de Turing

Linguagens Formais e Autômatos

CEFET

Dezembro 2025

Sumário

1 Exercício 1: Construção de Máquinas de Turing	2
1.1 1.a) $L = \{w \mid w \in \{a, b, c\}^* \text{ e } w \text{ começa com } ab\}$	2
1.2 1.b) $L = \{a^n b^n c^n \mid n \geq 0\}$	3
1.3 1.c) $L = \{a^n b^m \mid n, m \geq 0 \text{ e } n = 2m\}$	4
1.4 1.d) $L = \{ww^R \mid w \in \{a, b\}^*\}$	5
1.5 1.e) $L = \{w \mid w \in \{a, b\}^* \text{ e } n_a(w) = n_b(w)\}$	6
1.6 1.f) $L = \{1^n 0^{n+3} \mid n \geq 0\}$	7
1.7 1.g) $L = \{a^n b^{2n} c^{n-1} \mid n > 0\}$	8
1.8 1.h) $L = \{a^i b^j a^k \mid j = \max(i, k)\}$	9
1.9 1.i) $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$	10
2 Exercício 2: Análise de Máquina de Turing	11
2.1 Diagrama de Estados da MT	11
2.2 Definição Formal da MT	11
2.3 Função de Transição (δ)	12
2.4 Questão a) Sequência de Configurações	13
2.5 Resumo da Questão a)	14
2.6 Questão b) Linguagem Aceita	15
3 Exercício 3: MT Multifita	16
4 Exercício 4: MT Não-Determinística	17

1 Exercício 1: Construção de Máquinas de Turing

Enunciado: Construa Máquinas de Turing padrão que aceitem as seguintes linguagens:

1.1 1.a) $L = \{w \mid w \in \{a, b, c\}^* \text{ e } w \text{ começa com } ab\}$

Descrição

Esta MT verifica se a cadeia de entrada começa com os símbolos ab em sequência.

Algoritmo

1. Verifica se o primeiro símbolo é a
2. Move para a direita e verifica se o segundo símbolo é b
3. Se ambos estiverem corretos, aceita
4. Caso contrário, rejeita

Diagrama de Estados

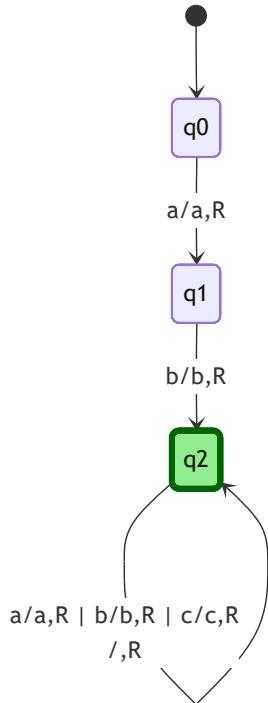


Figura 1: MT para $L = \{w \mid w \text{ começa com } ab\}$

1.2 1.b) $L = \{a^n b^n c^n \mid n \geq 0\}$

Descrição

Esta MT aceita cadeias com igual quantidade de a 's, b 's e c 's em sequência.

Algoritmo

1. Marca um a com X
2. Busca e marca um b com Y
3. Busca e marca um c com Z
4. Retorna ao início e repete até não haver mais símbolos não marcados
5. Se sobrar algum símbolo não pareado, rejeita

Diagrama de Estados

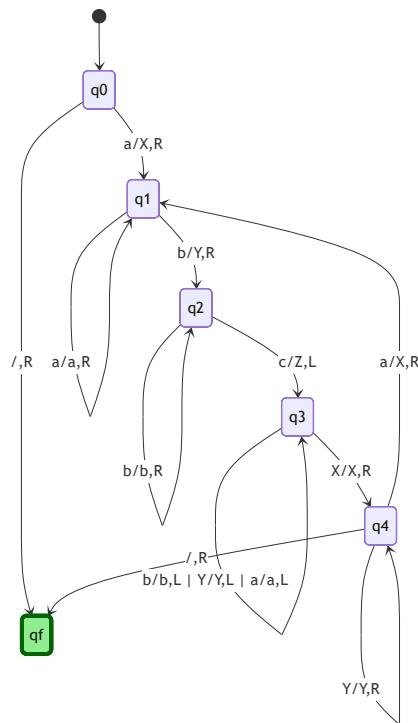


Figura 2: MT para $L = \{a^n b^n c^n \mid n \geq 0\}$

1.3 1.c) $L = \{a^n b^m \mid n, m \geq 0 \text{ e } n = 2m\}$

Descrição

Esta MT aceita cadeias onde o número de a 's é o dobro do número de b 's.

Algoritmo

1. Marca dois a 's com X
2. Busca e marca um b com Y
3. Retorna e repete
4. Aceita se todos os símbolos forem pareados corretamente

Diagrama de Estados

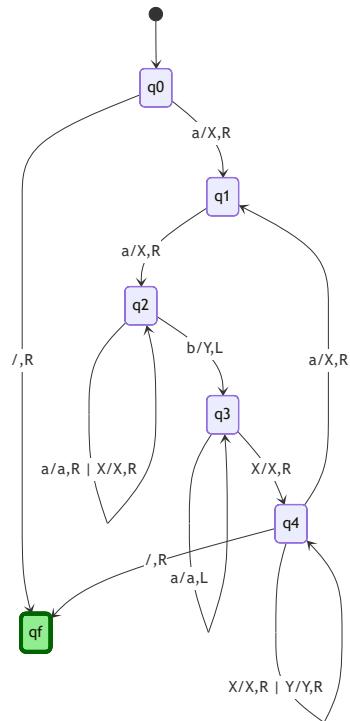


Figura 3: MT para $L = \{a^n b^m \mid n, m \geq 0 \text{ e } n = 2m\}$

1.4 1.d) $L = \{ww^R \mid w \in \{a, b\}^*\}$

Descrição

Esta MT aceita palíndromos pares, ou seja, cadeias que são iguais lidas da esquerda para a direita e da direita para a esquerda.

Algoritmo

1. Memoriza e marca o primeiro símbolo
2. Vai até o último símbolo não marcado
3. Verifica se é igual ao memorizado
4. Marca e retorna ao início
5. Repete até todos estarem marcados

Diagrama de Estados

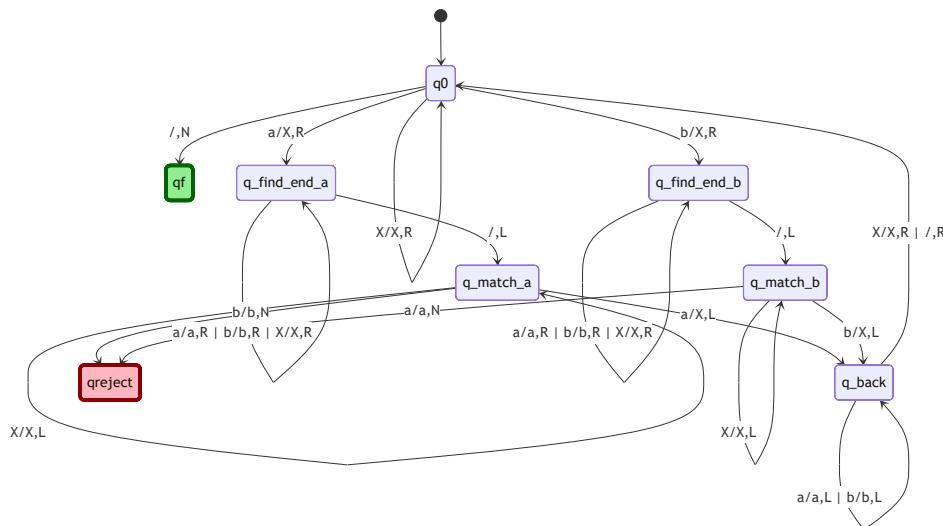


Figura 4: MT para $L = \{ww^R\}$ (palíndromos pares)

1.5 1.e) $L = \{w \mid w \in \{a, b\}^* \text{ e } n_a(w) = n_b(w)\}$

Descrição

Esta MT aceita cadeias com igual quantidade de a 's e b 's, em qualquer ordem.

Algoritmo

1. Encontra um a não marcado, marca com X
2. Busca um b não marcado, marca com X
3. (Ou vice-versa: encontra b primeiro, depois a)
4. Retorna ao início e repete
5. Aceita se todos os símbolos forem pareados

Diagrama de Estados

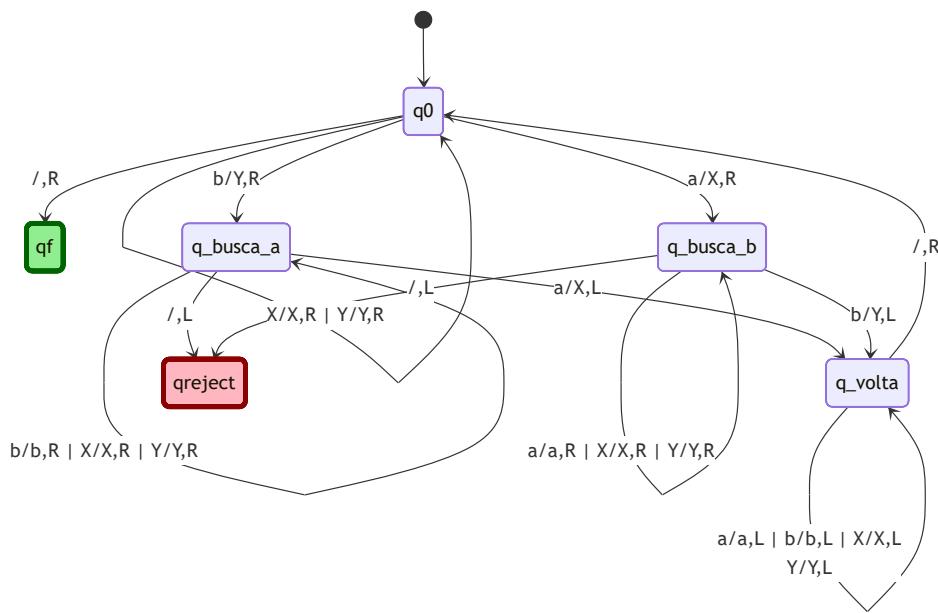


Figura 5: MT para $L = \{w \mid n_a(w) = n_b(w)\}$

1.6 1.f) $L = \{1^n 0^{n+3} \mid n \geq 0\}$

Descrição

Esta MT aceita cadeias com n uns seguidos de $n + 3$ zeros.

Algoritmo

1. Verifica que existem pelo menos 3 zeros no início (após os 1's)
2. Para cada 1, marca e busca um 0 correspondente
3. Aceita se a contagem estiver correta

Diagrama de Estados

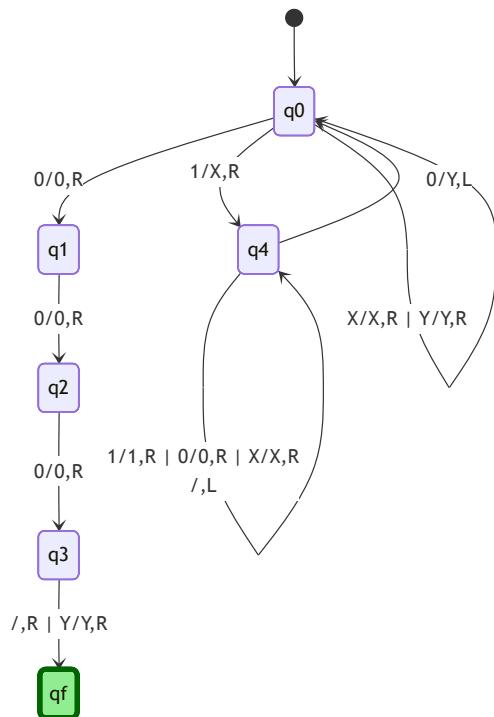


Figura 6: MT para $L = \{1^n 0^{n+3}\}$

1.7 1.g) $L = \{a^n b^{2n} c^{n-1} \mid n > 0\}$

Descrição

Esta MT aceita cadeias onde:

- Número de b 's é o dobro do número de a 's
- Número de c 's é um a menos que o número de a 's

Algoritmo

1. Para cada a : marca dois b 's e um c (exceto o primeiro a que não marca c)
2. Verifica que a contagem está correta

Diagrama de Estados

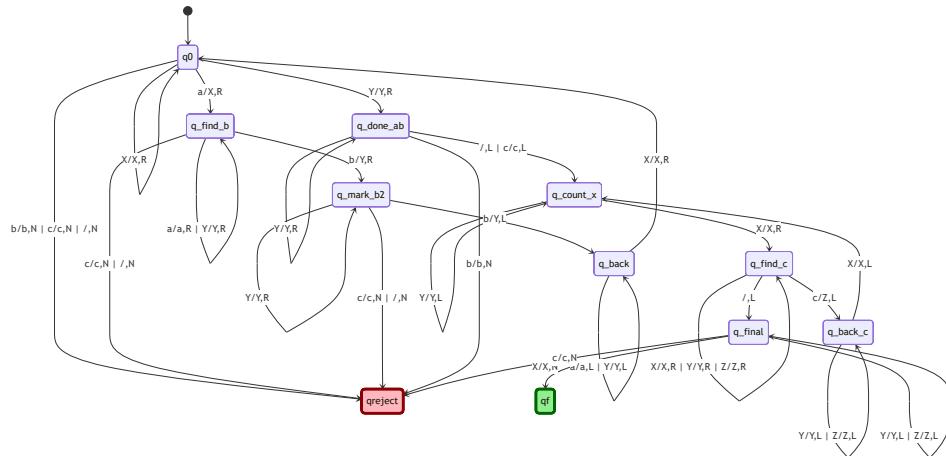


Figura 7: MT para $L = \{a^n b^{2n} c^{n-1}\}$

1.8 1.h) $L = \{a^i b^j a^k \mid j = \max(i, k)\}$

Descrição

Esta MT aceita cadeias onde o número de b 's é igual ao máximo entre o número de a 's à esquerda e à direita.

Algoritmo

1. Pareia a 's da esquerda com a 's da direita
2. O excedente (se houver) deve ser igual aos b 's
3. Os b 's pareados com a 's dos dois lados devem bater

Diagrama de Estados

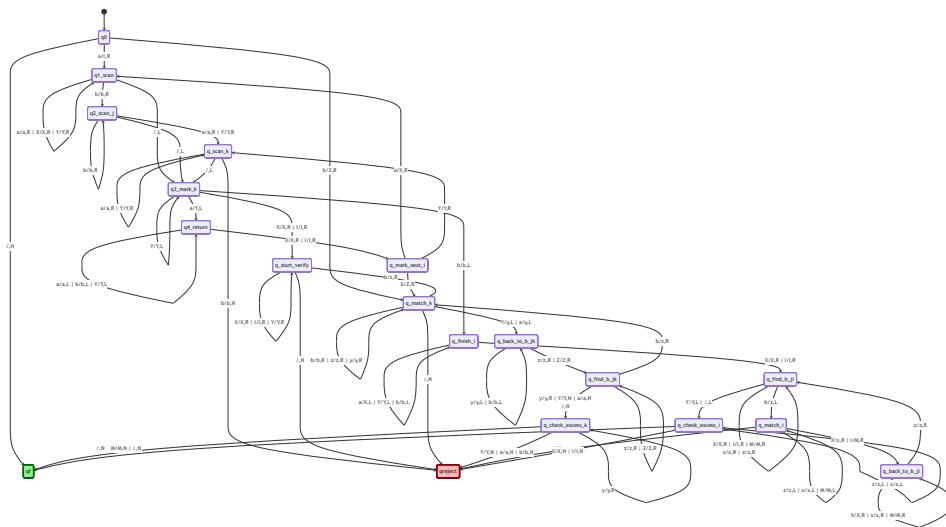


Figura 8: MT para $L = \{a^i b^j a^k \mid j = \max(i, k)\}$

1.9 1.i) $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$

Descrição

Esta MT aceita cadeias onde o número de b 's é igual ao número de a 's à esquerda **OU** ao número de a 's à direita.

Algoritmo

A MT verifica ambas as condições:

1. **Caso $i = j$:** Pareia cada a da esquerda com um b
2. **Caso $j = k$:** Pareia cada b com um a da direita
3. Aceita se qualquer uma das condições for satisfeita

Complexidade

- **Estados:** 27
- **Transições:** 116

Diagrama de Estados

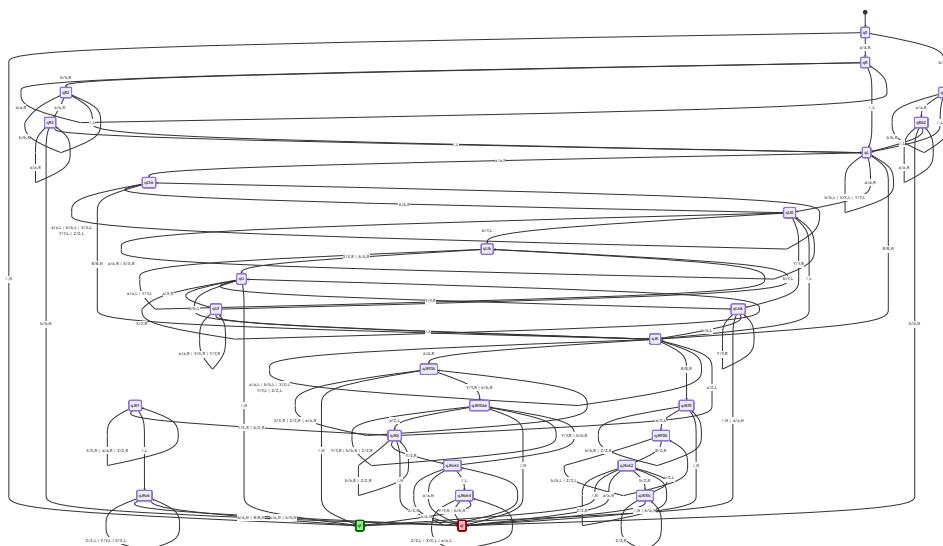


Figura 9: MT para $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$

Versão sem Estado de Rejeição Explícito

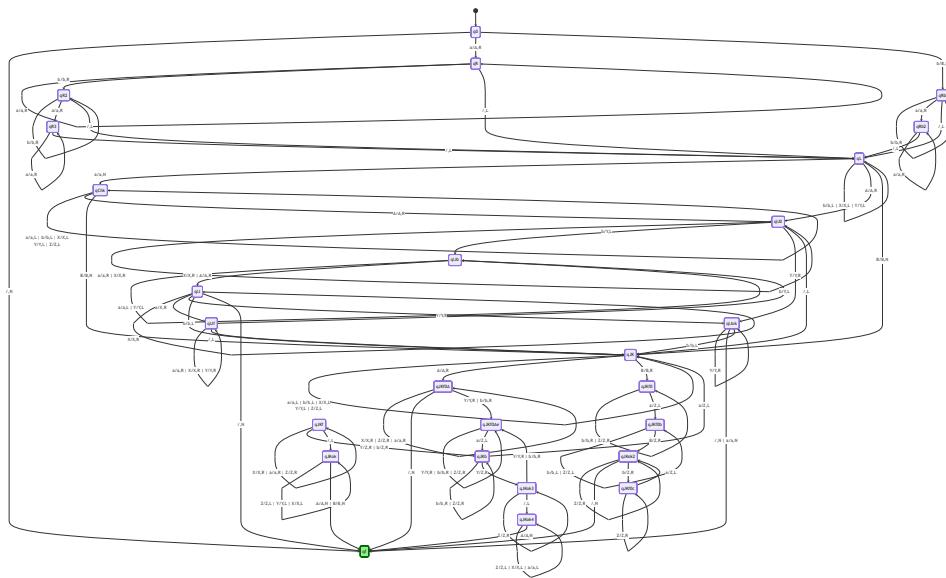


Figura 10: MT para $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$ (sem transições para q_r)

2 Exercício 2: Análise de Máquina de Turing

Enunciado: Dada a MT da figura abaixo, sobre o alfabeto $\Sigma = \{a, b\}$, responda:

O símbolo $\#$ não pertence ao alfabeto e foi inserido somente para marcar o início da fita.

2.1 Diagrama de Estados da MT

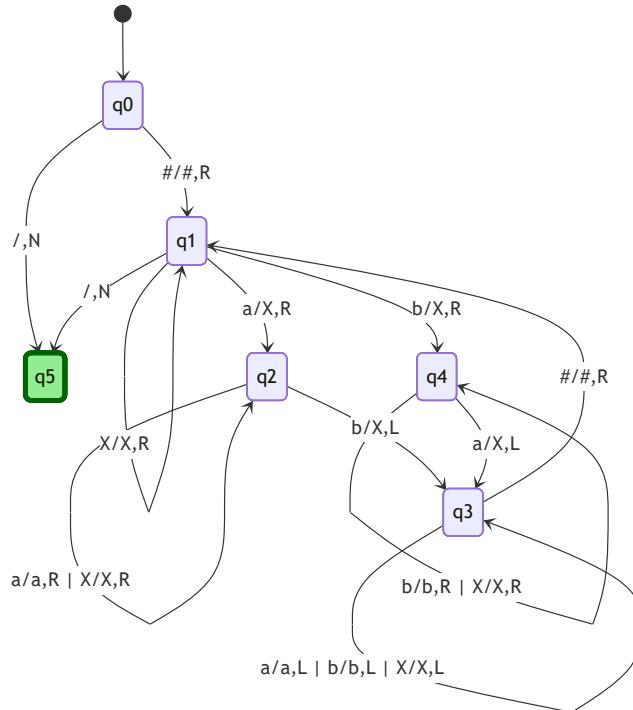


Figura 11: Máquina de Turing para $L = \{w \mid n_a(w) = n_b(w)\}$

2.2 Definição Formal da MT

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{aceita}, q_{rejeita})$$

Componente	Valor
Estados (Q)	$\{q_0, q_1, q_2, q_3, q_4, q_5, q_r\}$
Alfabeto de Entrada (Σ)	$\{a, b\}$
Alfabeto da Fita (Γ)	$\{a, b, \#, X, _\}$
Símbolo Branco	—
Estado Inicial	q_0
Estado de Aceitação	q_5
Estado de Rejeição	q_r (implícito)

Tabela 1: Componentes da Máquina de Turing

Notação Formal das Transições

A função de transição $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$ é definida pelas seguintes regras:

$$\begin{aligned} \delta(q_0, \#) &= (q_1, \#, R) & (1) \\ \delta(q_0, _) &= (q_5, _, N) & (2) \\ \delta(q_1, a) &= (q_2, X, R) & (3) \\ \delta(q_1, b) &= (q_4, X, R) & (4) \\ \delta(q_1, X) &= (q_1, X, R) & (5) \\ \delta(q_1, _) &= (q_5, _, N) & (6) \\ \delta(q_2, a) &= (q_2, a, R) & (7) \\ \delta(q_2, X) &= (q_2, X, R) & (8) \\ \delta(q_2, b) &= (q_3, X, L) & (9) \\ \delta(q_3, a) &= (q_3, a, L) & (10) \\ \delta(q_3, b) &= (q_3, b, L) & (11) \\ \delta(q_3, X) &= (q_3, X, L) & (12) \\ \delta(q_3, \#) &= (q_1, \#, R) & (13) \\ \delta(q_4, b) &= (q_4, b, R) & (14) \\ \delta(q_4, X) &= (q_4, X, R) & (15) \\ \delta(q_4, a) &= (q_3, X, L) & (16) \end{aligned}$$

2.3 Função de Transição (δ)

Estado	Lê	Próximo	Escreve	Move	Descrição
q_0	#	q_1	#	R	Início: Move para direita
q_0	_	q_5	_	N	String vazia: aceita
q_1	a	q_2	X	R	Marca a e busca b
q_1	b	q_4	X	R	Marca b e busca a
q_1	X	q_1	X	R	Pula marcados
q_1	_	q_5	_	N	Todos pareados: aceita
q_2	a	q_2	a	R	Pula a's
q_2	X	q_2	X	R	Pula X's
q_2	b	q_3	X	L	Encontrou b: marca e retorna
q_3	a	q_3	a	L	Retorna pulando a
q_3	b	q_3	b	L	Retorna pulando b
q_3	X	q_3	X	L	Retorna pulando X
q_3	#	q_1	#	R	Chegou ao início
q_4	b	q_4	b	R	Pula b's
q_4	X	q_4	X	R	Pula X's
q_4	a	q_3	X	L	Encontrou a: marca e retorna

Tabela 2: Função de Transição (16 transições)

2.4 Questão a) Sequência de Configurações

Cadeia: #abab

ACEITA

Resultado: ACEITA (2 a's e 2 b's pareados)

```

Passo 0: q0, [#]abab    → (q0, #) → (q1, #, R)
Passo 1: q1, #[a]bab    → (q1, a) → (q2, X, R)
Passo 2: q2, #X[b]ab    → (q2, b) → (q3, X, L)
Passo 3: q3, #[X]Xab    → (q3, X) → (q3, X, L)
Passo 4: q3, #[#]XXab   → (q3, #) → (q1, #, R)
Passo 5: q1, #[X]Xab    → (q1, X) → (q1, X, R)
Passo 6: q1, #X[X]ab    → (q1, X) → (q1, X, R)
Passo 7: q1, #XX[a]b     → (q1, a) → (q2, X, R)
Passo 8: q2, #XXX[b]     → (q2, b) → (q3, X, L)
...
... (retorno para #)
Passo 17: q1, #XXXX[_]   → (q1, _) → (q5, _, N)

```

Cadeia: #aabba

REJEITA

Resultado: REJEITA (3 a's e 2 b's - sobrou 1 a sem par)

```

Passo 0: q0, [#]aabba   → (q0, #) → (q1, #, R)
Passo 1: q1, #[a]abba   → (q1, a) → (q2, X, R)
Passo 2: q2, #X[a]bba   → (q2, a) → (q2, a, R)
Passo 3: q2, #Xa[b]ba   → (q2, b) → (q3, X, L)
...
... (pareamento de a-b, a-b)
Passo 19: q1, #XXXX[a]   → (q1, a) → (q2, X, R)
Passo 20: q2, #XXXXX[_]  → SEM TRANSIÇÃO!

```

Cadeia: #bbabaa

ACEITA

Resultado: ACEITA (3 a's e 3 b's pareados)

```

Passo 0: q0, [#]bbabaa  → (q0, #) → (q1, #, R)
Passo 1: q1, #[b]babaa  → (q1, b) → (q4, X, R)
Passo 2: q4, #X[b]abaa  → (q4, b) → (q4, b, R)
Passo 3: q4, #Xb[a]baa  → (q4, a) → (q3, X, L)
...
... (pareamento de b-a, b-a, b-a)
Passo 35: q1, #XXXXXX[_] → (q1, _) → (q5, _, N)

```

Cadeia: #abaabab

REJEITA

Resultado: REJEITA (4 a's e 3 b's - sobrou 1 a sem par)

Passo 0: q0, [#]abaabab → (q0, #) → (q1, #, R)

Passo 1: q1, #[a]baabab → (q1, a) → (q2, X, R)

Passo 2: q2, #X[b]aabab → (q2, b) → (q3, X, L)

... (pareamento de a-b, a-b, a-b)

Passo 34: q1, #XXXXXX[a]X → (q1, a) → (q2, X, R)

Passo 35: q2, #XXXXXX[X] → (q2, X) → (q2, X, R)

Passo 36: q2, #XXXXXXXX[_] → SEM TRANSIÇÃO!

2.5 Resumo da Questão a)

Cadeia	# de a's	# de b's	Resultado
#abab	2	2	ACEITA
#aabba	3	2	REJEITA
#bbabaa	3	3	ACEITA
#abaabab	4	3	REJEITA

Tabela 3: Resultados para as cadeias da questão a)

2.6 Questão b) Linguagem Aceita

Resposta

A linguagem L aceita por essa Máquina de Turing é:

$$L = \{w \in \{a, b\}^* \mid n_a(w) = n_b(w)\}$$

Onde:

- $n_a(w)$ = número de ocorrências de a na cadeia w
- $n_b(w)$ = número de ocorrências de b na cadeia w

Descrição

A MT aceita todas as cadeias sobre $\{a, b\}$ que contêm **exatamente o mesmo número de a 's e b 's**, independente da ordem em que aparecem.

Exemplos de cadeias aceitas

- ϵ (cadeia vazia)
- ab, ba
- $aabb, abab, abba, baab, baba, bbaa$
- $aaabbb, ababab, aababb$, etc.

Exemplos de cadeias rejeitadas

- a, b
- aab, abb, aaa, bbb
- $aabba, abbba$, etc.

Funcionamento do Algoritmo

1. **Início** (q_0): Lê o marcador $\#$ e vai para q_1
2. **Busca de par** (q_1):
 - Se encontra a : marca com X , vai para q_2 (buscar b)
 - Se encontra b : marca com X , vai para q_4 (buscar a)
 - Se encontra $_$: todos pareados \rightarrow ACEITA
3. **Busca de b** (q_2): Avança até encontrar b , marca e retorna
4. **Busca de a** (q_4): Avança até encontrar a , marca e retorna
5. **Retorno** (q_3): Volta ao início para próximo par
6. **Aceitação** (q_5): Estado final

3 Exercício 3: MT Multifita

Enunciado: Refaça os exercícios usando MT multifita com uma complexidade de tempo inferior à MT padrão.

A ser implementado...

4 Exercício 4: MT Não-Determinística

Enunciado: Refaça os exercícios usando MT Não-Determinísticas com uma complexidade de tempo inferior à MT padrão.

A ser implementado...

Anexo: Comandos CLI para Testes

Os arquivos JSON das Máquinas de Turing podem ser testados usando o CLI:

```
# Testar exercício 1.a
node cli.js --def input/MT_exe1_a.json --test "ab,abc,abcc,ba,a"

# Testar exercício 1.i
node cli.js --def input/MT_exe1_i.json --test "ab,aab,abb,aabb" --verbose

# Testar exercício 2 (cadeias da questão a)
node cli.js --def input/MT_exe2.json --test "#abab,#aabba,#bbabaa,#abaabab"

# Teste com saída detalhada
node cli.js --def input/MT_exe2.json --test "#abab" --verbose
```