Lista de Exercícios

Máquinas de Turing

Linguagens Formais e Autômatos

CEFET

Dezembro 2025

Sumário

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

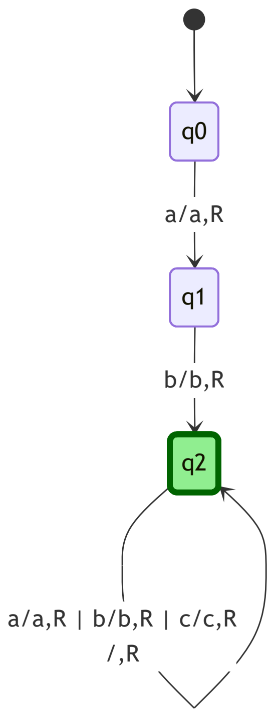

Figura 1: MT para $L = \{w \mid w \text{ começa com } ab\}$

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

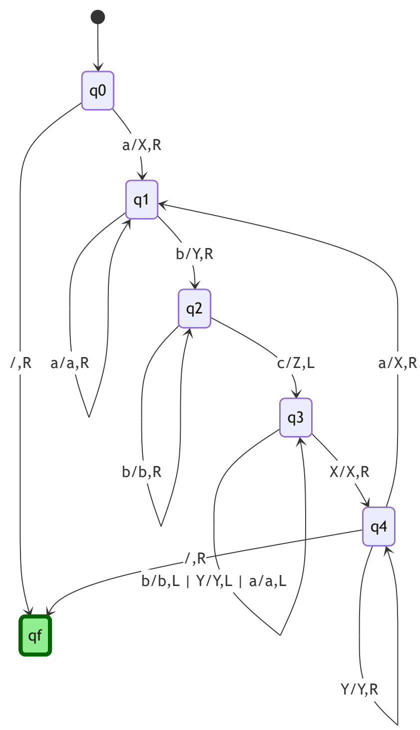

Figura 2: MT para $L = \{a^n b^n c^n \mid n \geq 0\}$

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

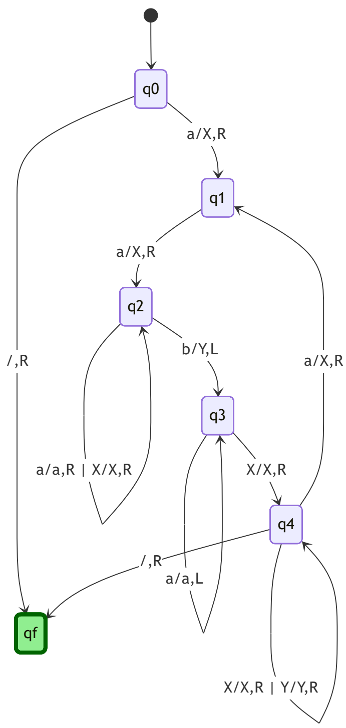

Figura 3: MT para $L = \{a^n b^m \mid n = 2m\}$

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

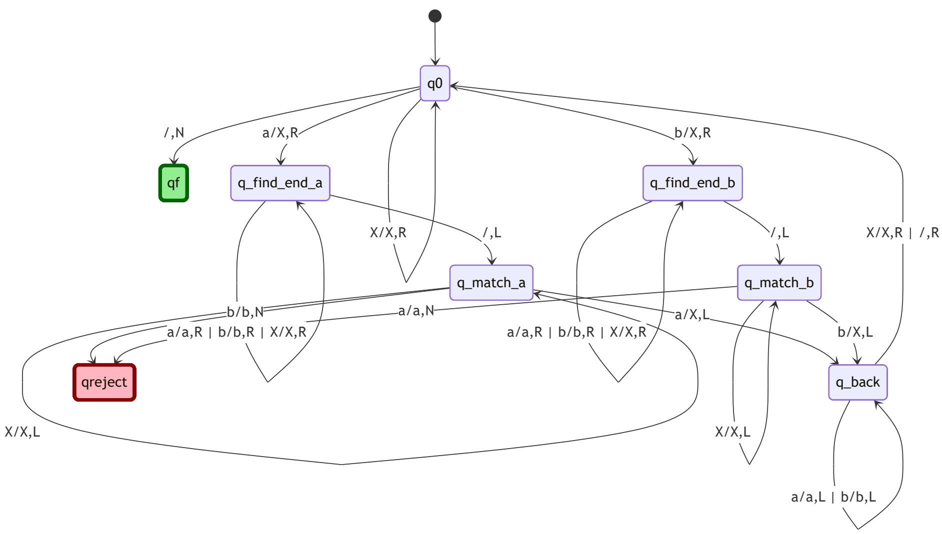

Figura 4: MT para $L = \{ww^R\}$ (palíndromos pares)

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

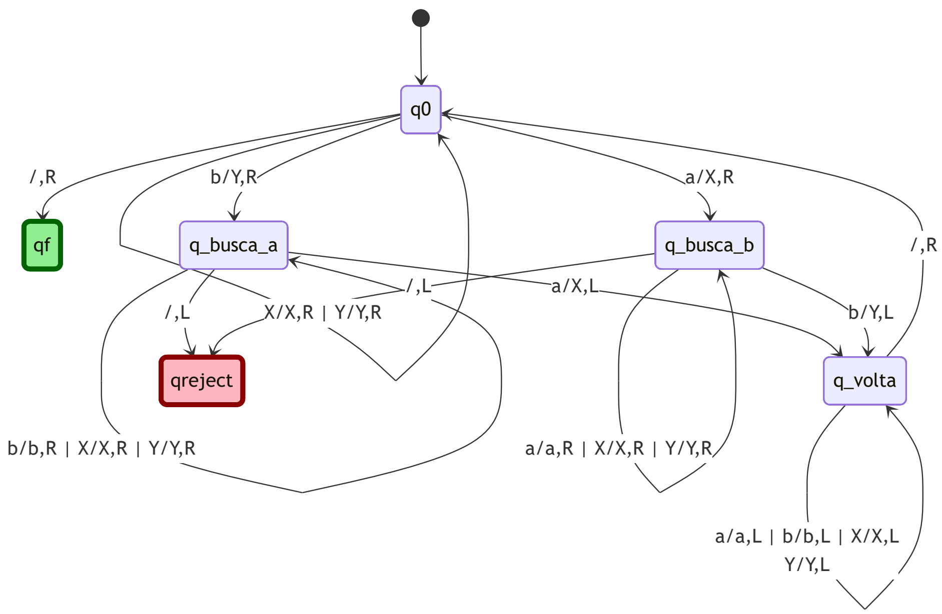

Figura 5: MT para $L = \{w \mid n_a(w) = n_b(w)\}$

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

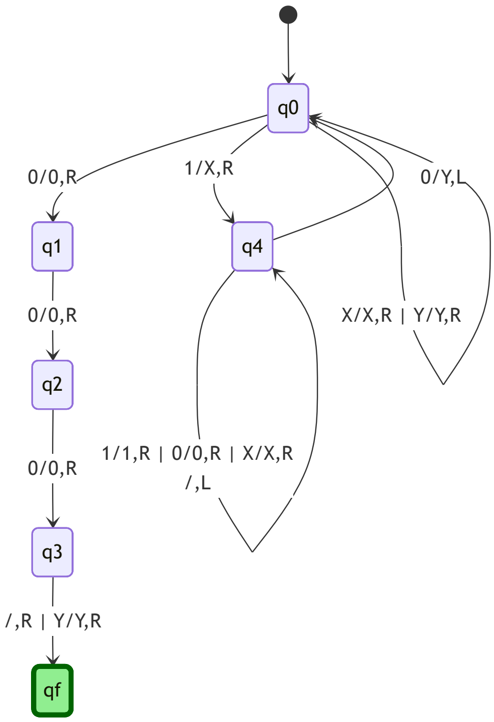

Figura 6: MT para $L = \{1^n 0^{n+3}\}$

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

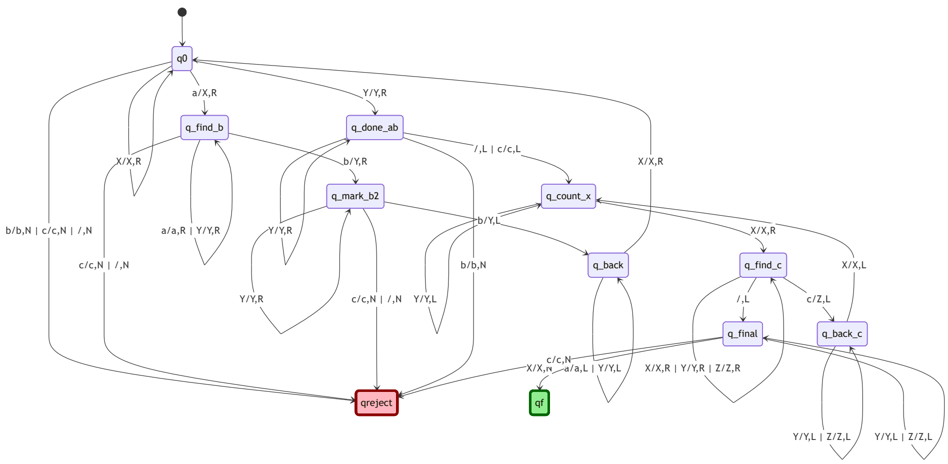

Figura 7: MT para $L = \{a^n b^{2n} c^{n-1}\}$

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

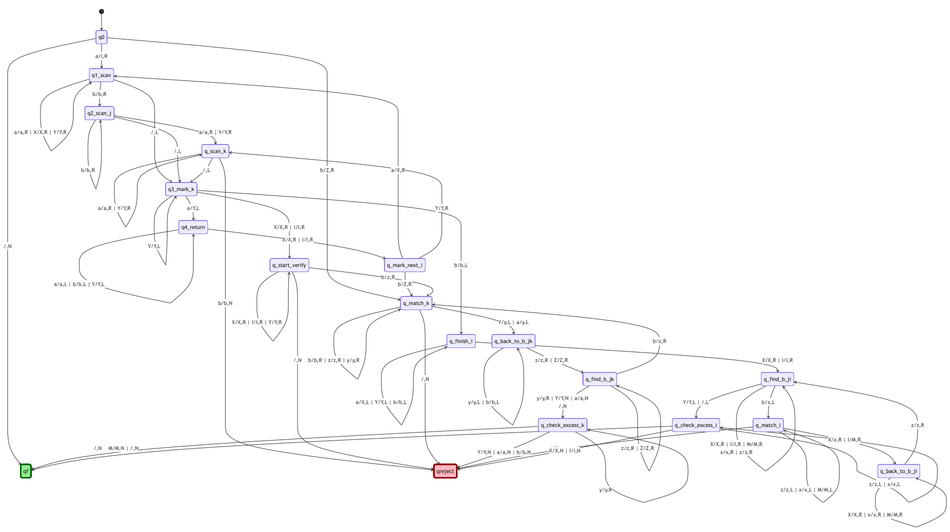

Figura 8: MT para $L = \{a^i b^j a^k \mid j = \max(i, k)\}$

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

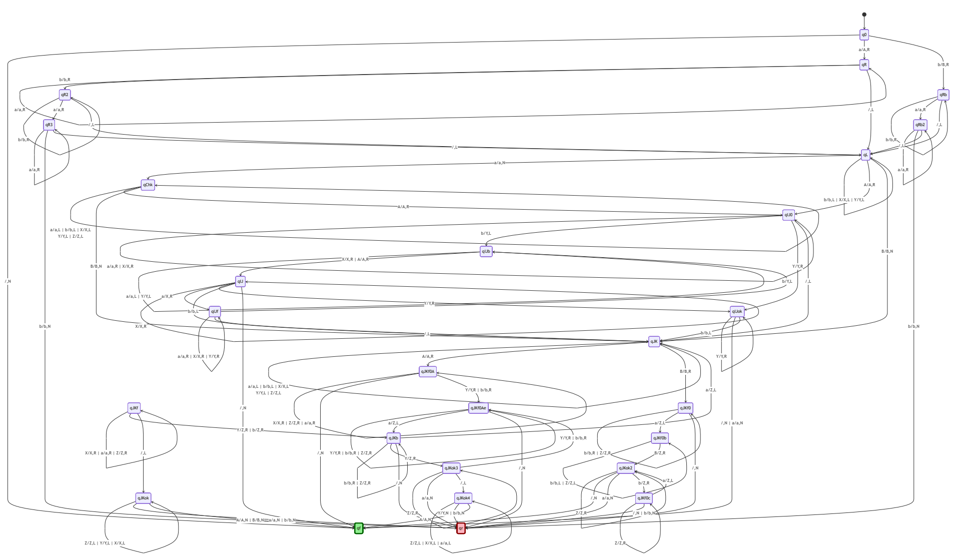

Figura 9: MT para $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$

Versão sem Estado de Rejeição Explícito

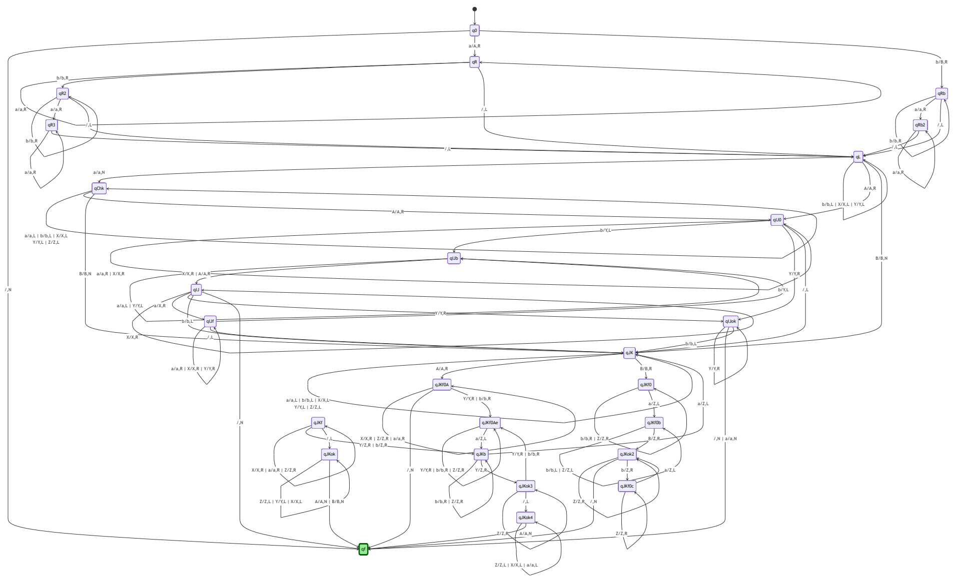

Figura 10: MT para $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$ (sem transições para q_r)

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

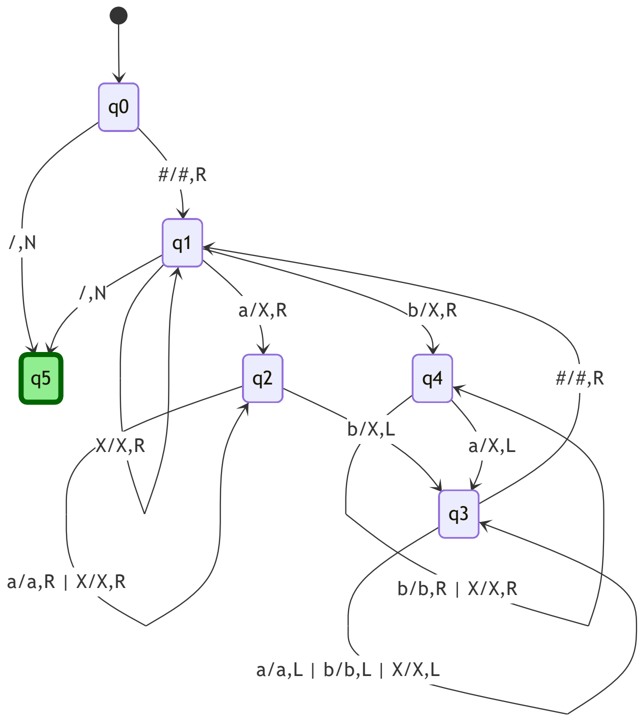

Figura 11: Máquina de Turing para $L = \{w \mid n_a(w) = n_b(w)\}$

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 $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$ é definida pelas seguintes regras:

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

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, #XXXXX[a]X → (q1, a) -> (q2, X, R)
Passo 35: q2, #XXXXXX[X] → (q2, X) -> (q2, X, R)
Passo 36: q2, #XXXXXXX[_] → 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:

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

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 → **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.

3.1 Introdução às MT Multifita

Uma **Máquina de Turing Multifita** possui k fitas, cada uma com seu próprio cabeçote de leitura/escrita. Em cada passo, a máquina:

1. Lê os símbolos sob todos os k cabeçotes simultaneamente
2. Dependendo do estado e dos símbolos lidos, transita para um novo estado
3. Escreve novos símbolos em cada fita e move cada cabeçote independentemente

Vantagem de Complexidade

Teorema

Uma MT multifita com k fitas pode simular qualquer computação que uma MT padrão faz em tempo $T(n)$ em tempo $O(T(n)^2)$. Porém, para muitos problemas, a MT multifita pode resolver em $O(n)$ o que a MT padrão resolve em $O(n^2)$.

A principal vantagem é usar fitas extras como **contadores** ou **buffers**, evitando múltiplas passagens pela entrada.

3.2 3.a) $L = \{w \mid w$ **começa com** $ab\}$

Análise de Complexidade

MT Padrão: $O(1)$ - Apenas verifica 2 símbolos
MT Multifita: Não há ganho - O problema já é trivial

Este problema não se beneficia de múltiplas fitas pois a verificação é feita em tempo constante.

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Para cada grupo de (a, b, c), percorre a fita inteira
MT Multifita (3 fitas): $O(n)$ - Uma única passagem

Algoritmo Multifita

1. **Fita 1:** Entrada original
2. **Fita 2:** Contador de a's (representação unária)
3. **Fita 3:** Contador de b's

```
Passo 1: Percorre a's na fita 1, escrevendo I na fita 2 para cada a
Passo 2: Percorre b's na fita 1:
         - Para cada b, remove um I da fita 2
         - Adiciona um I na fita 3
Passo 3: Percorre c's na fita 1:
         - Para cada c, remove um I da fita 3
Passo 4: Se fita 2 e fita 3 estão vazias → ACEITA
```

Transições (notação: [fita1, fita2, fita3])

$$
\begin{gathered}
(q_0, [a, \_, \_]) \to (q_0, [a, I, \_], [R, R, N]) \\
(q_0, [b, I, \_]) \to (q_1, [b, \_, I], [R, L, R]) \\
(q_1, [b, I, \_]) \to (q_1, [b, \_, I], [R, L, R]) \\
(q_1, [c, \_, I]) \to (q_2, [c, \_, \_], [R, N, L]) \\
(q_2, [c, \_, I]) \to (q_2, [c, \_, \_], [R, N, L]) \\
(q_2, [\_, \_, \_]) \to (q_f, [\_, \_, \_], [N, N, N])
\end{gathered}
$$

Diagrama de Estados

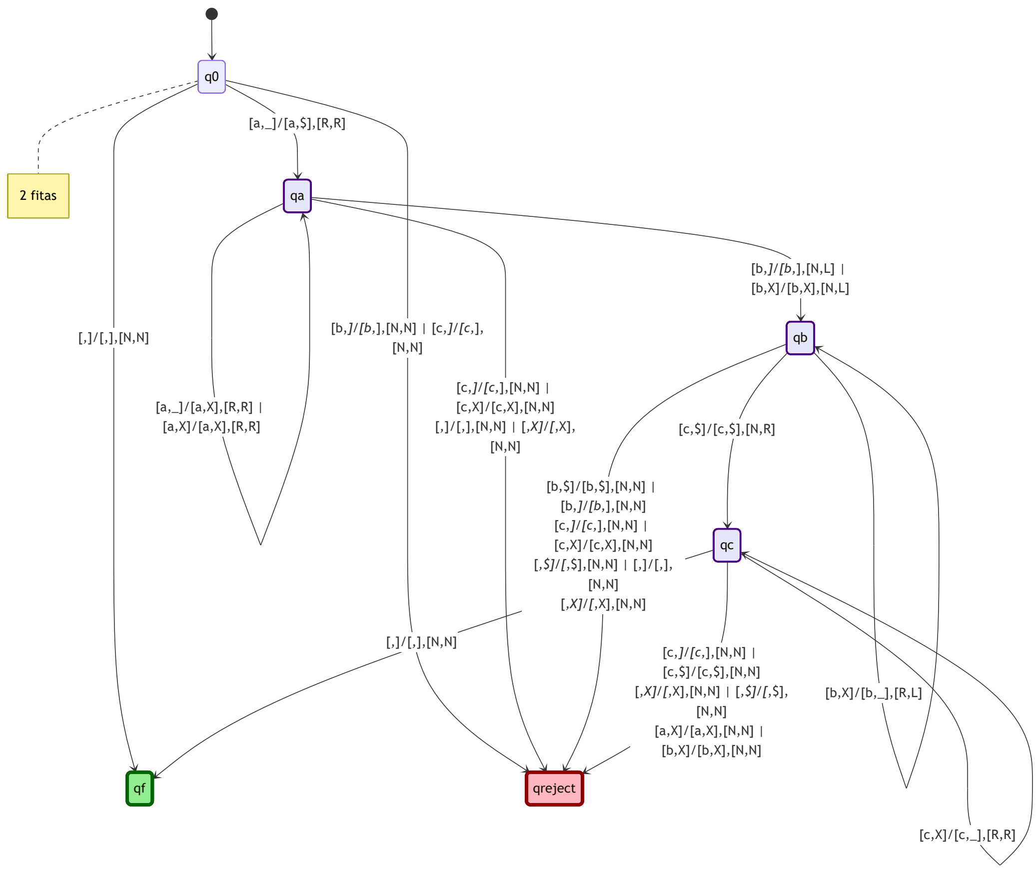

Figura 12: MT Multifita (3 fitas) para $L = \{a^n b^n c^n\}$

3.4 3.c) $L = \{a^n b^m \mid n = 2m\}$

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Marca 2 a's e 1 b por passagem
MT Multifita (2 fitas): $O(n)$ - Conta a's, desconta 2 por b

Algoritmo Multifita

1. **Fita 1:** Entrada
2. **Fita 2:** Contador de a's

```
Passo 1: Para cada 'a' na fita 1, escreve 'I' na fita 2
Passo 2: Para cada 'b' na fita 1, apaga DOIS 'I' da fita 2
Passo 3: Se fita 2 vazia ao final → ACEITA
```

Exemplo: $aaaaabb$ **(aceita:** $n = 4$, $m = 2$, $4 = 2 \times 2$**)**

```
Fita 1: [a]aaaabb    Fita 2: [_]
Fita 1: a[a]aabb     Fita 2: I[_]
Fita 1: aa[a]abb     Fita 2: II[_]
Fita 1: aaa[a]bb     Fita 2: III[_]
Fita 1: aaaa[b]b     Fita 2: IIII[_] → apaga 2: II[_]
Fita 1: aaaab[b]     Fita 2: II[_] → apaga 2: [_]
Fita 1: aaaaabb[_]   Fita 2: [_] → ACEITA
```

Diagrama de Estados

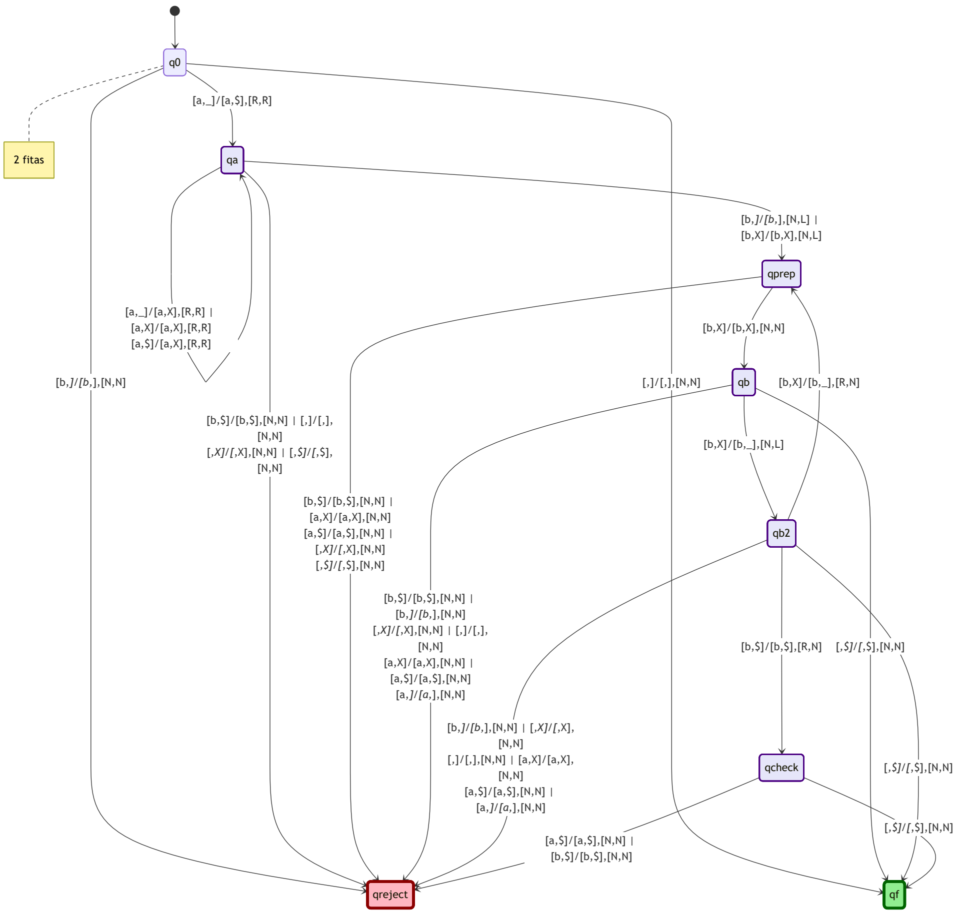

Figura 13: MT Multifita (2 fitas) para $L = \{a^n b^m \mid n = 2m\}$

3.5 3.d) $L = \{ww^R \mid w \in \{a,b\}^*\}$ (Palíndromos pares)

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Compara primeiro com último, volta ao início
MT Multifita (2 fitas): $O(n)$ - Copia e compara em paralelo

Algoritmo Multifita

1. **Fita 1:** Entrada (lida da esquerda para direita)
2. **Fita 2:** Cópia da entrada (lida da direita para esquerda)

```
Passo 1: Copia toda a entrada da fita 1 para a fita 2
Passo 2: Move cabeçote da fita 1 para o início
Passo 3: Move cabeçote da fita 2 para o fim
Passo 4: Compara símbolo por símbolo:
         - Fita 1 avança (→), Fita 2 retrocede (←)
         - Se todos iguais → ACEITA
```

Exemplo: $abba$ **(aceita:** $w = ab$, $w^R = ba$**)**

```
Após cópia:
Fita 1: [a]bba_    (leitura →)
Fita 2: _abba[_]   (leitura ←, posiciona no fim)

Comparação:
Fita 1: [a]bba    Fita 2: abb[a] → a=a
Fita 1: a[b]ba    Fita 2: ab[b]a → b=b
Fita 1: ab[b]a    Fita 2: a[b]ba → b=b
Fita 1: abb[a]    Fita 2: [a]bba → a=a
ACEITA!
```

Diagrama de Estados

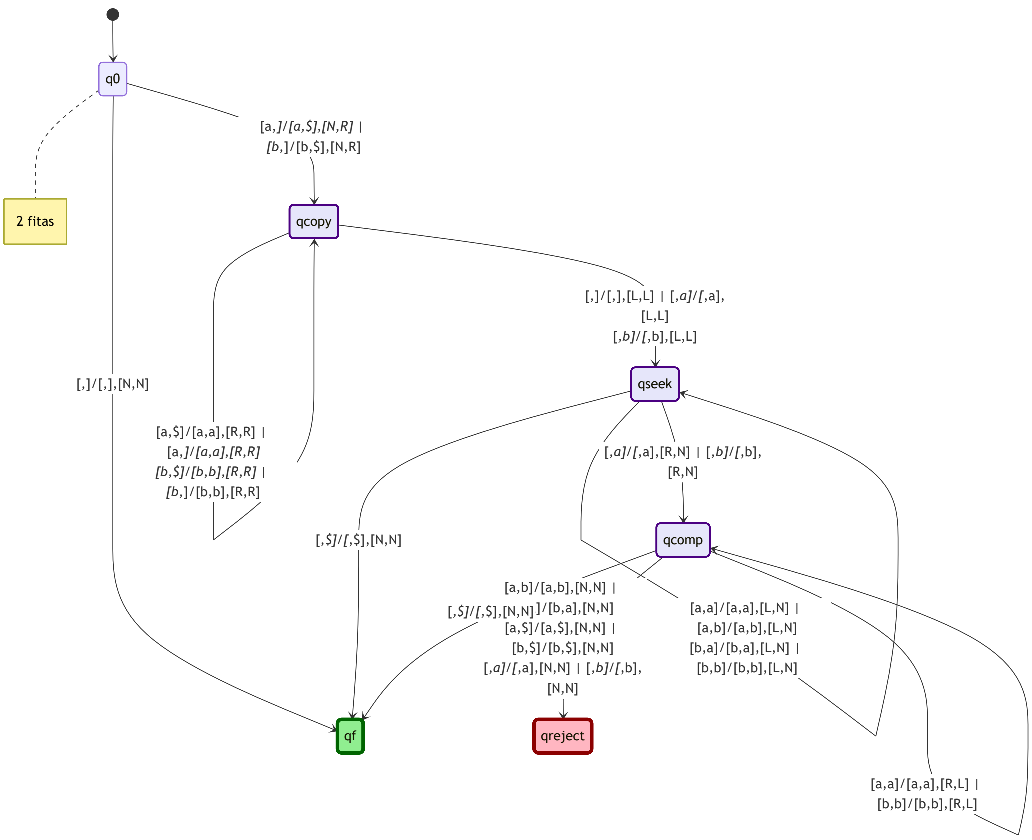

Figura 14: MT Multifita (2 fitas) para $L = \{ww^R\}$

3.6 3.e) $L = \{w \mid n_a(w) = n_b(w)\}$

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Pareia cada a com um b, volta ao início
MT Multifita (2 fitas): $O(n)$ - Usa contador na fita 2

Algoritmo Multifita

1. **Fita 1:** Entrada
2. **Fita 2:** Contador (diferença entre a's e b's)

```
Para cada símbolo na fita 1:
  - Se 'a': incrementa contador (escreve I na fita 2)
  - Se 'b': decrementa contador (apaga I da fita 2)
    - Se contador já vazio, marca como negativo
No final: aceita se contador = 0
```

Exemplo: *abba* **(aceita: 2** a**'s e 2** b**'s)**

```
Fita 1: [a]bba    Fita 2: [_] → escreve I → [I]
Fita 1: a[b]ba    Fita 2: [I] → apaga I → [_]
Fita 1: ab[b]a    Fita 2: [_] → marca negativo → [-]
Fita 1: abb[a]    Fita 2: [-] → cancela negativo → [_]
Fita 1: abba[_]   Fita 2: [_] → contador = 0 → ACEITA
```

Diagrama de Estados

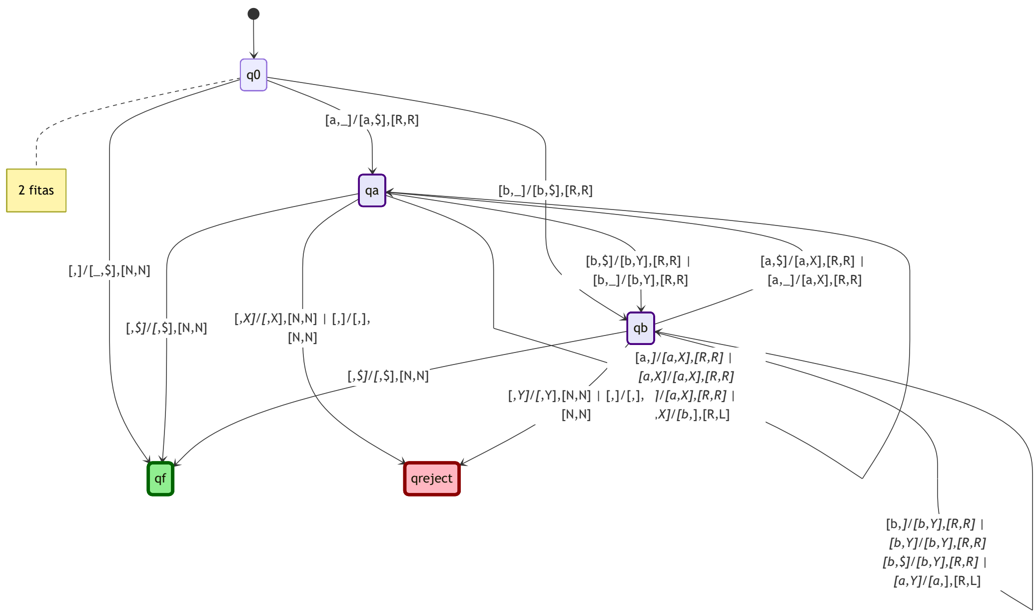

Figura 15: MT Multifita (2 fitas) para $L = \{w \mid n_a(w) = n_b(w)\}$

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Marca 1's e 0's em múltiplas passagens
MT Multifita (2 fitas): $O(n)$ - Conta 1's, pula 3 zeros, compara

Algoritmo Multifita

```
Passo 1: Para cada '1' na fita 1, escreve 'I' na fita 2
Passo 2: Lê exatamente 3 zeros (os "+3" obrigatórios)
Passo 3: Para cada '0' restante, apaga um 'I' da fita 2
Passo 4: Se fita 2 vazia → ACEITA
```

Exemplo: 110000 **(aceita:** $n = 2$, $2 + 3 = 5$**... ops, rejeitaria)**

Exemplo: 11000000 **(aceita:** $n = 2$, 0's$= 5 = 2 + 3$)

```
Fita 1: [1]1000000   Fita 2: [_] → I
Fita 1: 1[1]000000   Fita 2: I[_] → II
Fita 1: 11[0]00000   Fita 2: II (pula 3 zeros)
Fita 1: 110[0]0000   Fita 2: II
Fita 1: 1100[0]000   Fita 2: II
Fita 1: 11000[0]00   Fita 2: I[I] → apaga → I
Fita 1: 110000[0]0   Fita 2: [I] → apaga → _
Fita 1: 1100000[_]   Fita 2: [_] → ACEITA
```

Diagrama de Estados

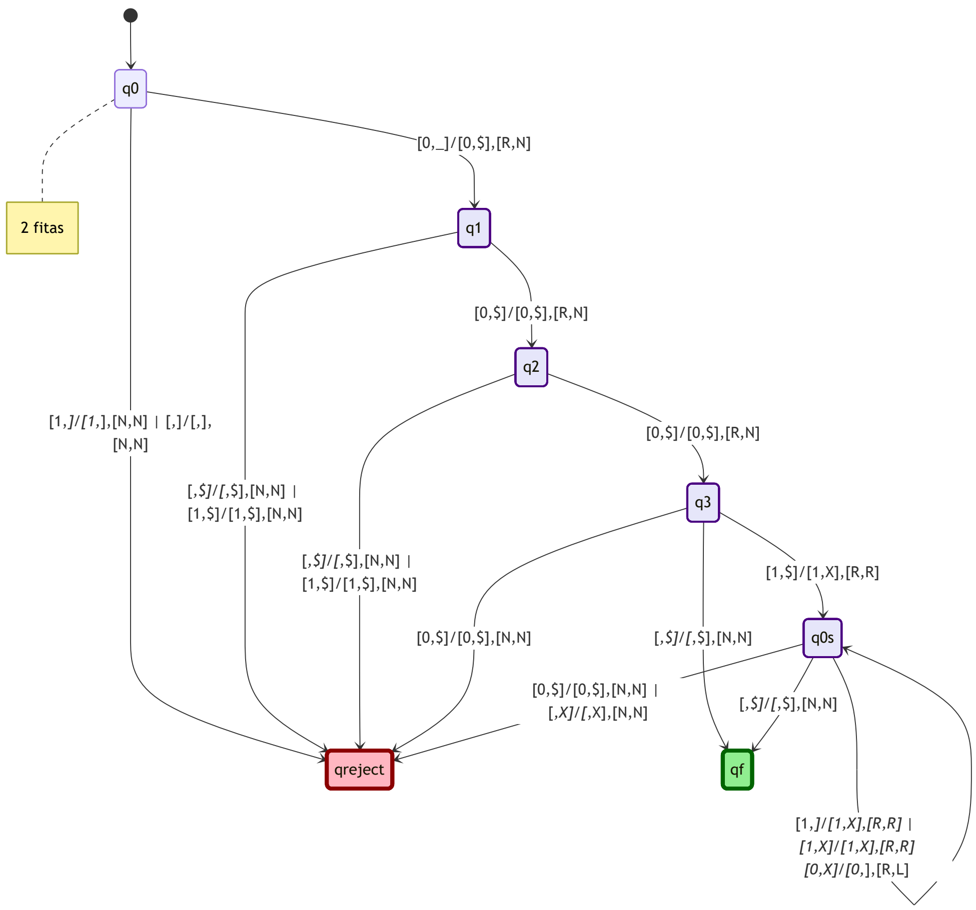

Figura 16: MT Multifita (2 fitas) para $L = \{1^n 0^{n+3}\}$

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Múltiplas passagens marcando símbolos
MT Multifita (2 fitas): $O(n)$ - Conta e verifica em uma passagem

Algoritmo Multifita

```
Passo 1: Para cada 'a', escreve 'I' na fita 2 (conta n)
Passo 2: Para cada 'b', apaga 0.5 'I' (ou seja, 2 b's = 1 I)
         Alternativa: para cada par de b's, apaga 1 'I'
Passo 3: Ao acabar b's, fita 2 deve estar vazia (confirma 2n)
Passo 4: Reconta os a's: n-1 deve ser igual ao número de c's
         - Pula primeiro c (é o "-1")
         - Para cada c seguinte, verifica correspondência
```

Exemplo: *abbbc* **(aceita:** $n = 2$, $b = 4 = 2 \times 2$, $c = 1 = 2 - 1$**)**

```
Fita 1: [a]abbbc    Fita 2: [_] → I
Fita 1: a[a]bbbc    Fita 2: I[_] → II
Fita 1: aa[b]bbc    Fita 2: II → par de b's → I
Fita 1: aab[b]bc    (continua par)
Fita 1: aabb[b]c    Fita 2: I → par de b's → _
Fita 1: aabbb[b]    (continua par)
Fita 1: aabbbb[c]   Fita 2: [_], primeiro c (grátis)
Fita 1: aabbbbc[_]  → ACEITA
```

Diagrama de Estados

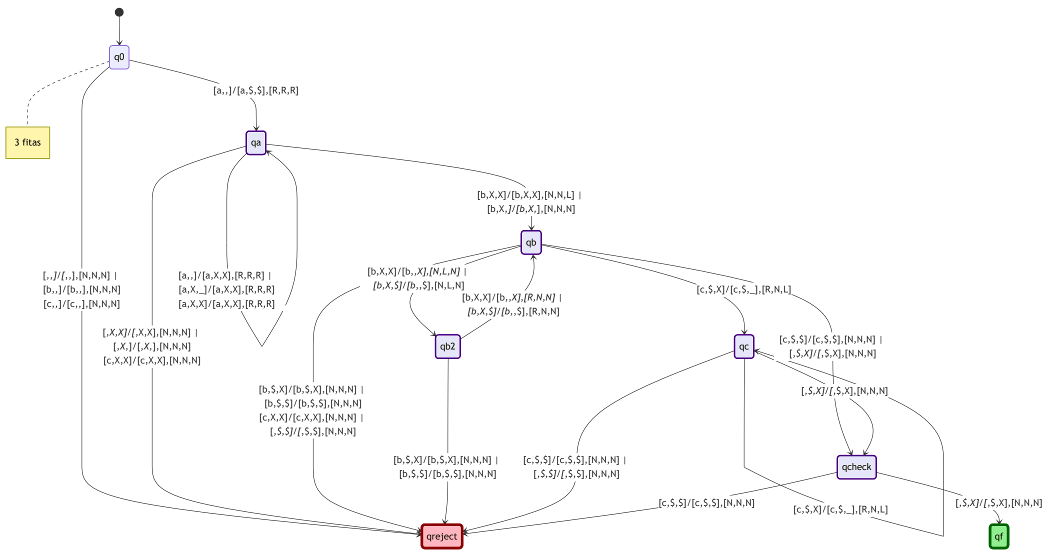

Figura 17: MT Multifita (2 fitas) para $L = \{a^n b^{2n} c^{n-1}\}$

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Múltiplas comparações
MT Multifita (3 fitas): $O(n)$ - Conta i, j, k e compara

Algoritmo Multifita

1. **Fita 1:** Entrada
2. **Fita 2:** Contador de i (a's à esquerda)
3. **Fita 3:** Contador de k (a's à direita)

```
Passo 1: Conta a's à esquerda na fita 2
Passo 2: Conta b's (guarda contagem j)
Passo 3: Conta a's à direita na fita 3
Passo 4: Determina max(i, k):
         - Compara fitas 2 e 3 símbolo a símbolo
         - O maior é max(i, k)
Passo 5: Verifica se j = max(i, k)
```

Exemplo: *aabbba* **(aceita:** $i = 2$**,** $j = 3$**,** $k = 1$**,** $\max(2, 1) = 2$**... rejeita)**

Exemplo: *aabba* **(aceita:** $i = 2$**,** $j = 2$**,** $k = 1$**,** $\max(2, 1) = 2 = j$**)**

```
Fita 1: [a]abba     Fita 2: [_]→I  Fita 3: [_]
Fita 1: a[a]bba     Fita 2: I[_]→II
Fita 1: aa[b]ba     (conta j=1)
Fita 1: aab[b]a     (conta j=2)
Fita 1: aabb[a]     Fita 3: [_]→I
Fita 1: aabba[_]
Comparação: Fita 2 = II (i=2), Fita 3 = I (k=1)
max(2,1) = 2 = j → ACEITA
```

Diagrama de Estados

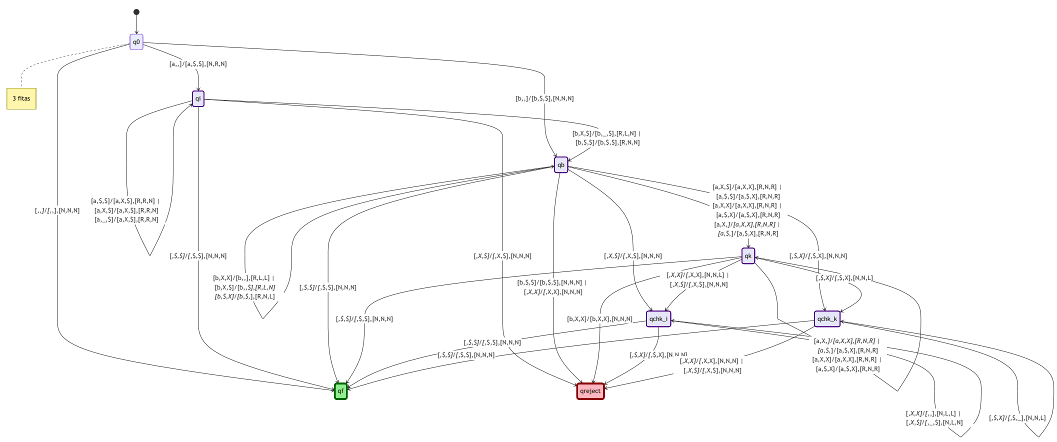

Figura 18: MT Multifita (3 fitas) para $L = \{a^i b^j a^k \mid j = \max(i,k)\}$

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ ou $O(n^3)$ - Tenta ambas condições
MT Multifita (3 fitas): $O(n)$ - Conta tudo e verifica condições

Algoritmo Multifita

1. **Fita 1:** Entrada
2. **Fita 2:** Contador de i
3. **Fita 3:** Contador de k

```
Passo 1: Conta a's à esquerda na fita 2 (i)
Passo 2: Conta b's, e simultaneamente:
         - Apaga um I da fita 2 para cada b
         - Se fita 2 esvazia com os b's → i = j
Passo 3: Conta a's à direita na fita 3 (k)
Passo 4: Compara j com k:
         - Se fita 3 tem exatamente j I's → j = k
Passo 5: ACEITA se qualquer condição for verdadeira
```

Exemplo: *aabba* ($i = 2$, $j = 2$, $k = 1$, $i = j$)

```
Fita 1: [a]abba     Fita 2: [_]→I
Fita 1: a[a]bba     Fita 2: I[_]→II
Fita 1: aa[b]ba     Fita 2: I[I]→I (apaga 1 para b)
Fita 1: aab[b]a     Fita 2: [I]→_ (apaga 1 para b)
Fita 2 vazia! → i = j confirmado → ACEITA
```

Exemplo: *abba* ($i = 1$, $j = 2$, $k = 1$, $j = k$**? Não,** $2 \neq 1$. $i = j$**? Não.)**

Rejeita.

Exemplo: *abbaa* ($i = 1$, $j = 2$, $k = 2$, $j = k$)

```
Fita 1: [a]bbaa     Fita 2: I
Fita 1: a[b]baa     Fita 2: _ (i j, continua)
Fita 1: ab[b]aa     Contador j = 2
Fita 1: abb[a]a     Fita 3: I
Fita 1: abba[a]     Fita 3: II
Comparação: j=2, Fita 3=II (k=2) → j = k → ACEITA
```

Diagrama de Estados

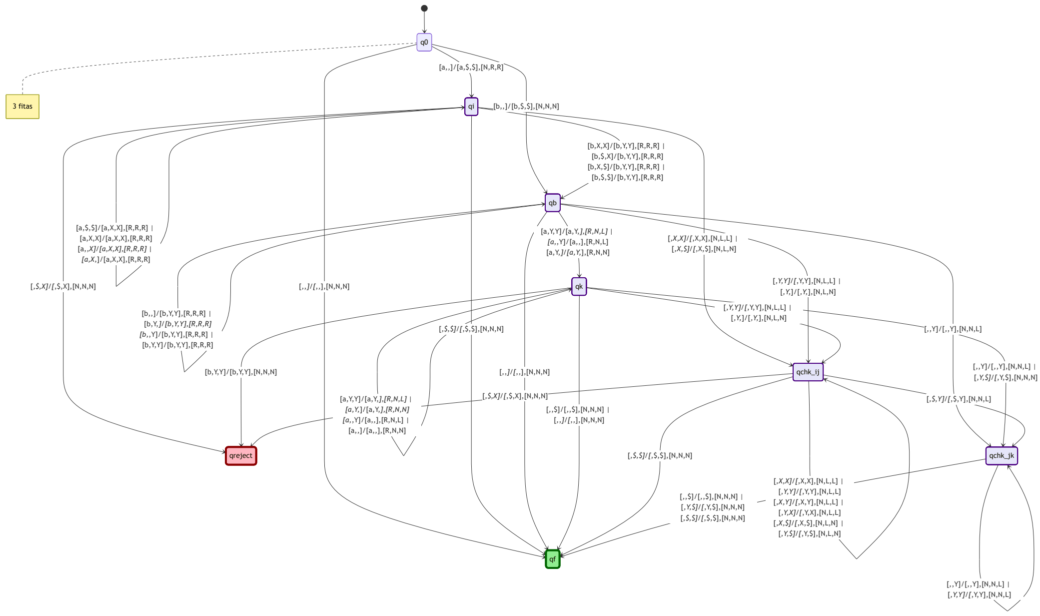

Figura 19: MT Multifita (3 fitas) para $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$

3.11 Resumo: Comparação de Complexidades

#	Linguagem	MT Padrão	MT Multifita	Fitas
a	w começa com ab	$O(1)$	$O(1)$	1
b	$a^n b^n c^n$	$O(n^2)$	$O(n)$	3
c	$a^n b^m, n = 2m$	$O(n^2)$	$O(n)$	2
d	ww^R	$O(n^2)$	$O(n)$	2
e	$n_a = n_b$	$O(n^2)$	$O(n)$	2
f	$1^n 0^{n+3}$	$O(n^2)$	$O(n)$	2
g	$a^n b^{2n} c^{n-1}$	$O(n^2)$	$O(n)$	2
h	$j = \max(i, k)$	$O(n^2)$	$O(n)$	3
i	$i = j$ ou $j = k$	$O(n^2)$	$O(n)$	3

Tabela 4: Comparação de complexidades entre MT padrão e MT multifita

Conclusão

A MT multifita reduz a complexidade de $O(n^2)$ para $O(n)$ na maioria dos problemas, utilizando fitas extras como **contadores** ou **buffers de comparação**. O princípio geral é:

- Evitar múltiplas passagens pela entrada
- Usar fitas auxiliares para armazenar contagens
- Processar a entrada em uma única varredura

4 Exercício 4: MT Não-Determinística

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

4.1 Introdução às MT Não-Determinísticas

Uma **Máquina de Turing Não-Determinística (MTND)** pode ter múltiplas transições possíveis para um mesmo par (estado, símbolo). Em cada passo, a máquina pode "escolher"qual transição seguir, explorando múltiplos caminhos simultaneamente.

Diferença Fundamental

Não-Determinismo

MT Determinística: Para cada (q, a), existe **no máximo uma** transição $\delta(q, a) = (q', b, D)$
MT Não-Determinística: Para cada (q, a), podem existir **múltiplas** transições:

$$\delta(q, a) \subseteq Q \times \Gamma \times \{L, R, N\}$$

A MTND **aceita** se **existe ao menos um caminho** que leva ao estado de aceitação.

Vantagem de Complexidade

Embora as MTNDs não sejam mais poderosas que MTs determinísticas (Teorema de Church-Turing), elas podem resolver certos problemas mais rapidamente:

- **"Adivinhar"soluções:** A MTND pode "adivinhar"não-deterministicamente onde está o meio de um palíndromo, quais elementos parear, etc.
- **Exploração paralela:** Conceptualmente, explora todos os caminhos em paralelo
- **Redução de complexidade:** Muitos problemas $O(n^2)$ podem ser resolvidos em $O(n)$ com não-determinismo

4.2 4.a) $L = \{w \mid w$ começa com $ab\}$

Análise de Complexidade

MT Padrão: $O(1)$ - Verifica 2 símbolos
MTND: Não há ganho - Problema já é trivial

Este exercício não se beneficia do não-determinismo pois a verificação já é feita em tempo constante.

Diagrama de Estados

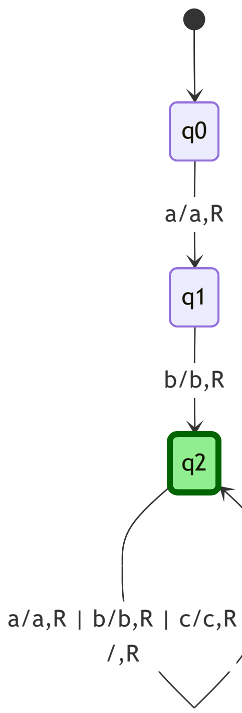

Figura 20: MTND para $L = \{w \mid w$ começa com $ab\}$ (idêntica à versão determinística)

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Múltiplas passagens marcando (a, b, c)
MTND: $O(n)$ - Adivinha não-deterministicamente quantos símbolos marcar

Estratégia Não-Determinística

A MTND **adivinha** quantos grupos (a, b, c) deve marcar por passagem, reduzindo o número de varreduras necessárias.

Algoritmo

1. Não-deterministicamente escolhe entre:
   - Continuar marcando grupos (a, b, c)
   - Parar e verificar se todos foram marcados
2. Para cada grupo: marca um a com X, um b com Y, um c com Z
3. Aceita se todos os símbolos estiverem marcados

Transições Não-Determinísticas

As seguintes transições ilustram o não-determinismo:

$$\delta(q_{\text{marca\_a}}, a) = \{(q_{\text{marca\_a}}, a, R), (q_{\text{busca\_b}}, X, R)\}$$
$$\delta(q_{\text{marca\_b}}, b) = \{(q_{\text{marca\_b}}, b, R), (q_{\text{busca\_c}}, Y, R)\}$$

Em $q_{\text{marca\_a}}$ ao ler a, a máquina pode:

- **Opção 1:** Continuar buscando ($a \to a$, move R)
- **Opção 2:** Marcar este a e buscar b correspondente ($a \to X$, vai para $q_{\text{busca\_b}}$)

Diagrama de Estados

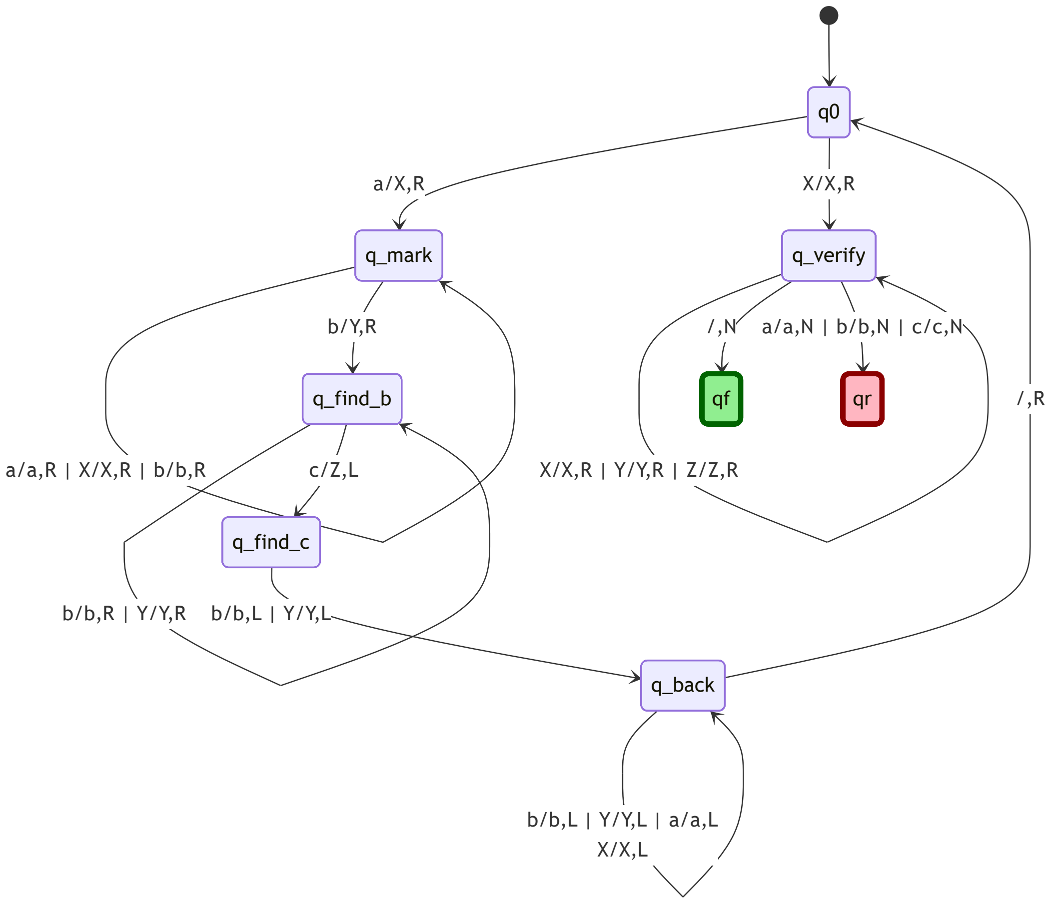

Figura 21: MTND para $L = \{a^n b^n c^n\}$ com 2 pares não-determinísticos

4.4 4.c) $L = \{a^{2m}b^m \mid m \geq 0\}$

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Marca 2 a's e 1 b por iteração
MTND: $O(n)$ - Adivinha não-deterministicamente qual b parear

Estratégia Não-Determinística

A MTND marca dois a's e então **adivinha** qual b deve ser pareado com este par.

Diagrama de Estados

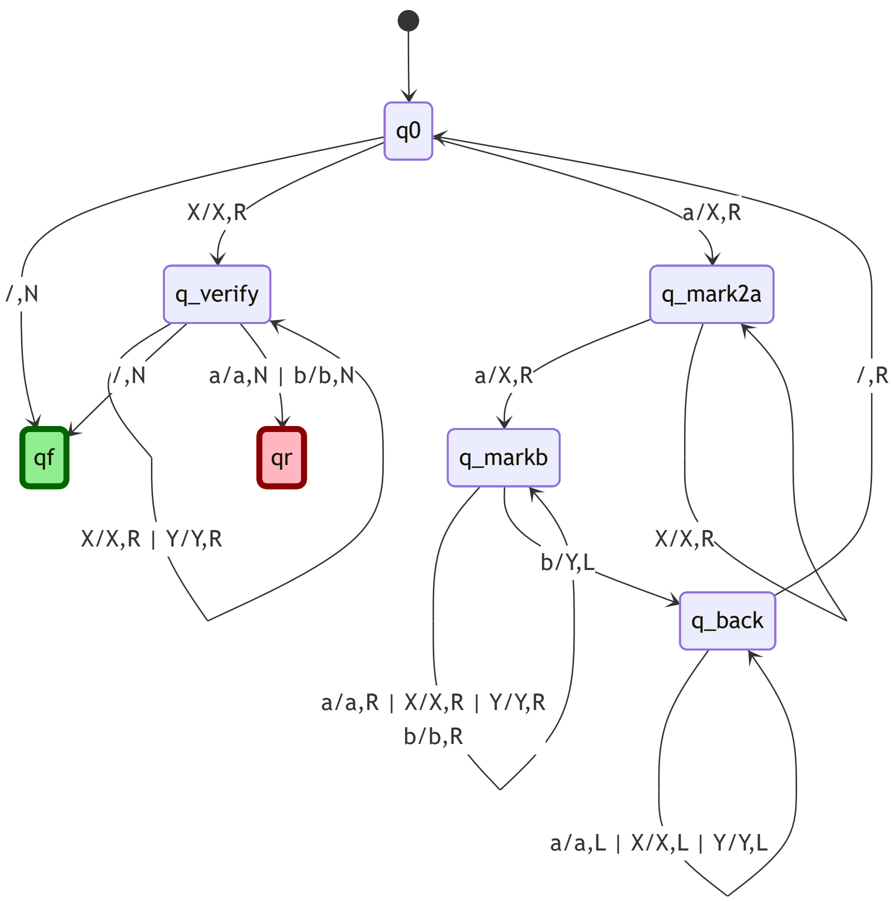

Figura 22: MTND para $L = \{a^{2m}b^m\}$ com 2 pares não-determinísticos

4.5 4.d) $L = \{ww^R \mid w \in \{a, b\}^*\}$ - Palíndromos Pares

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Compara primeiro com último, segundo com penúltimo, etc.
MTND: $O(n)$ - Adivinha não-deterministicamente onde está o meio

Estratégia Não-Determinística

A MTND **adivinha** onde está o ponto de divisão entre w e w^R. Se adivinhar corretamente, verifica a igualdade em tempo linear.

Algoritmo

1. Marca o primeiro símbolo não marcado com X (se a) ou Y (se b)
2. Avança pela fita
3. **Não-deterministicamente** escolhe entre:
   - Continuar avançando
   - **Parar aqui** (adivinhar que este é o meio/fim da metade direita)
4. Volta marcando o último símbolo não-marcado que deve corresponder ao primeiro
5. Verifica se corresponde (mesmo tipo: a com a, b com b)
6. Repete até todos estarem marcados

Transições Não-Determinísticas

$$\delta(q_{\text{scan}}, a) = \{(q_{\text{scan}}, a, R), (q_{\text{mark\_end}}, a, L)\}$$
$$\delta(q_{\text{scan}}, b) = \{(q_{\text{scan}}, b, R), (q_{\text{mark\_end}}, b, L)\}$$

Em q_{scan}, ao ler a ou b, a máquina pode:

- **Opção 1:** Continuar avançando (ainda não chegou ao meio)
- **Opção 2:** Parar e voltar (adivinhou que este é o ponto certo)

Exemplos

- **Aceita:** ϵ, aa, bb, $abba$, $aabbaa$, $baab$
- **Rejeita:** a, b, ab, aba, abc, aab, $babba$

Diagrama de Estados

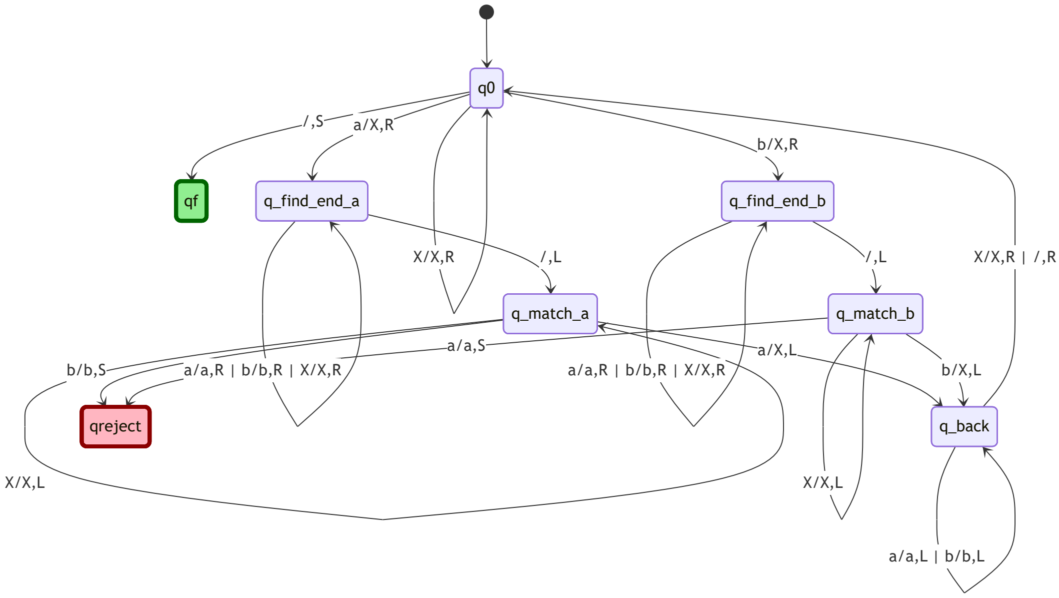

Figura 23: MTND para $L = \{ww^R\}$ com 2 pares não-determinísticos em q_{scan}

4.6 4.e) $L = \{w \mid w \in \{a, b\}^* \text{ e } \#a = \#b\}$

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Busca linear por pares a-b
MTND: $O(n)$ - Adivinha não-deterministicamente qual b parear com cada a

Estratégia Não-Determinística

A MTND, ao encontrar um a, **adivinha** qual b não-marcado deve ser pareado com ele, reduzindo múltiplas buscas.

Diagrama de Estados

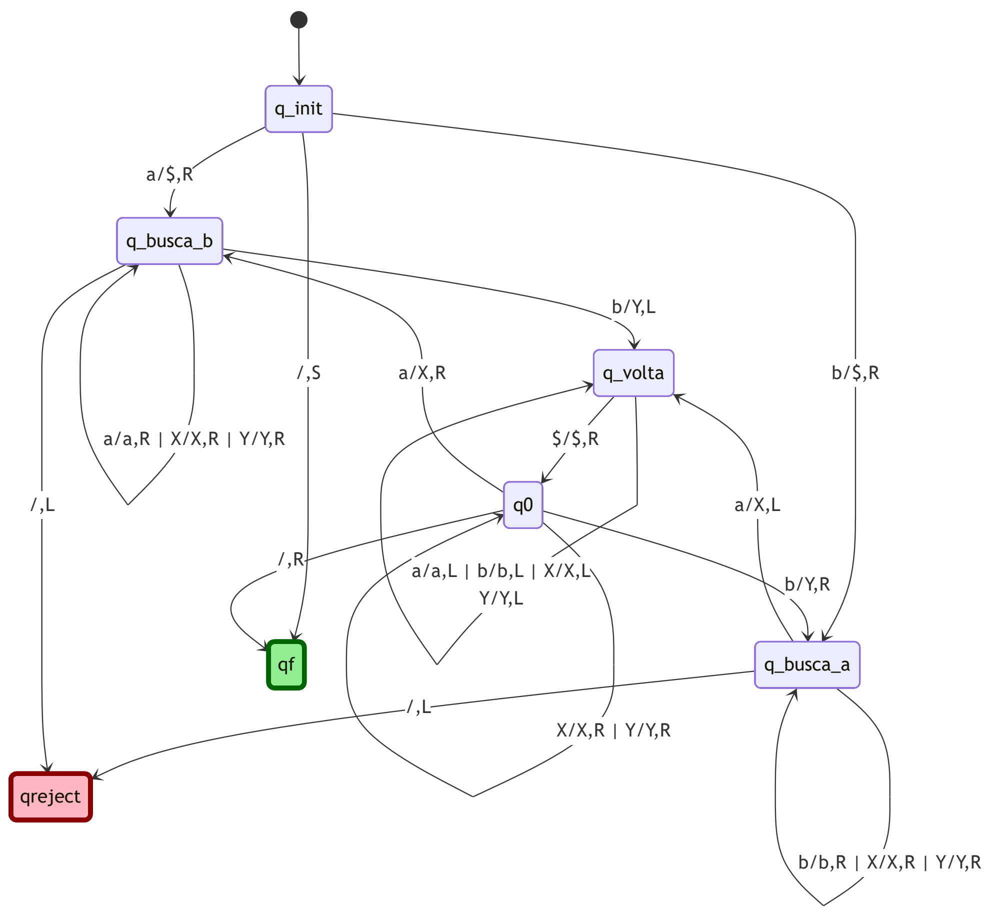

Figura 24: MTND para $L = \{w \mid \#a = \#b\}$ com 2 pares não-determinísticos

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Pareia cada 1 com um 0 (exceto 3)
MTND: $O(n)$ - Adivinha não-deterministicamente o valor de n

Estratégia Não-Determinística

A MTND **adivinha** quantos 1's existem e verifica se há exatamente $n + 3$ zeros.

Diagrama de Estados

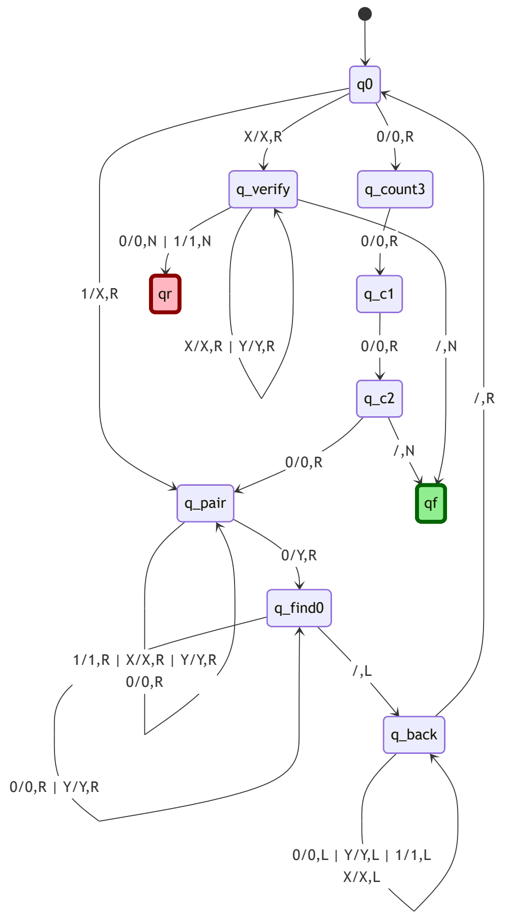

Figura 25: MTND para $L = \{1^n 0^{n+3}\}$ com 2 pares não-determinísticos

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Marca a, dois b's, um c por iteração
MTND: Determinística - Estrutura fixa não se beneficia do não-determinismo

Este problema mantém complexidade similar pois a estrutura $a^n b^{2n} c^{n-1}$ exige contagem precisa em sequência.

Diagrama de Estados

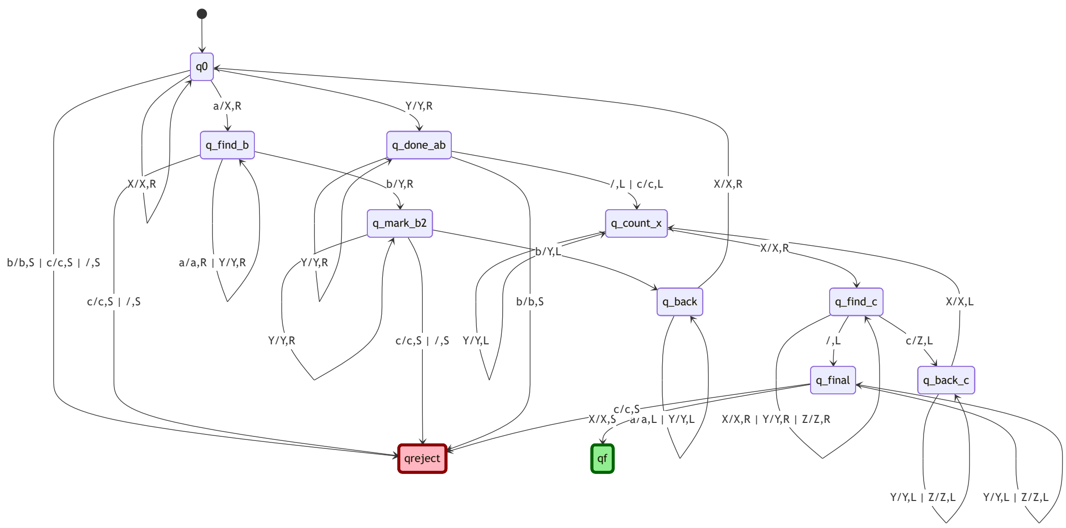

Figura 26: MTND para $L = \{a^n b^{2n} c^{n-1}\}$ (determinística)

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Compara i e k, depois verifica j
MTND: $O(n)$ - Adivinha qual é o máximo (i ou k) e verifica

Estratégia Não-Determinística

A MTND **adivinha** se $i > k$, $i < k$, ou $i = k$, e verifica se j corresponde ao máximo.

Algoritmo

1. Não-deterministicamente escolhe uma das hipóteses:
   - $i > k$: Verifica se $j = i$
   - $i < k$: Verifica se $j = k$
   - $i = k$: Verifica se $j = i = k$
2. Para a hipótese escolhida, faz pareamento linear

Diagrama de Estados

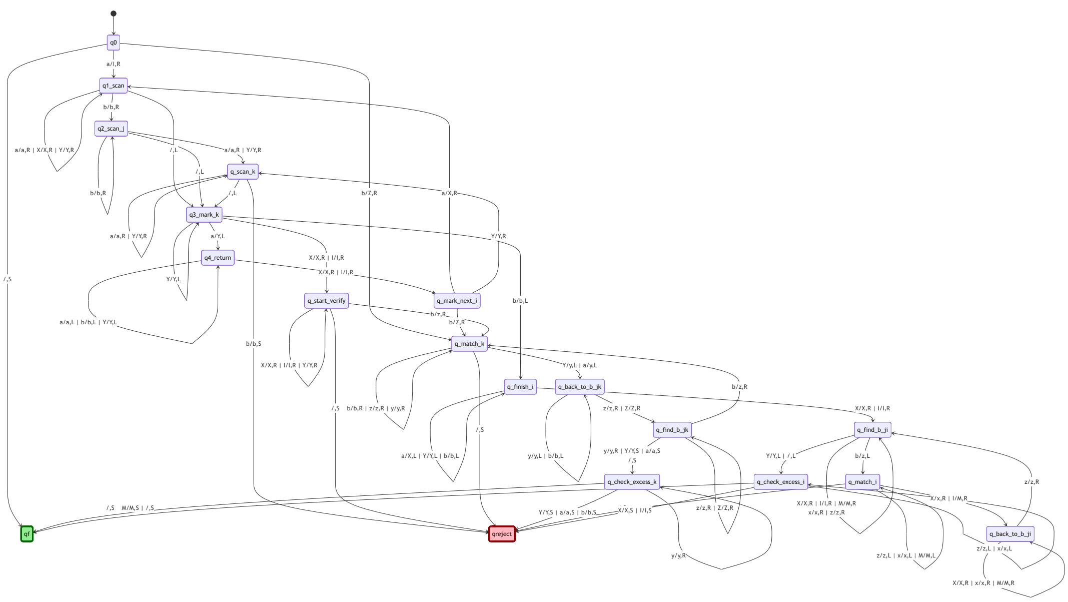

Figura 27: MTND para $L = \{a^i b^j a^k \mid j = \max(i, k)\}$ com 9 pares não-determinísticos

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

Análise de Complexidade

Comparação

MT Padrão: $O(n^2)$ - Testa ambas condições sequencialmente
MTND: $O(n)$ - Adivinha qual condição testar ($i = j$ ou $j = k$)

Estratégia Não-Determinística

A MTND **adivinha** qual das duas condições é verdadeira:

- **Ramo 1:** Testa se $i = j$ (ignora k)
- **Ramo 2:** Testa se $j = k$ (ignora i)

Se algum dos ramos aceitar, a cadeia é aceita.

Algoritmo

1. Não-deterministicamente escolhe testar:
   - **Condição 1 ($i = j$):** Pareia cada a à esquerda com um b, ignora a's à direita
   - **Condição 2 ($j = k$):** Pareia cada b com um a à direita, ignora a's à esquerda
2. Verifica se o pareamento escolhido é exato

Exemplos

- **Aceita:**
  - ab (1=1, teste $i = j$)
  - aab (2=1? não, mas 1=1 teste $j = k$)
  - aba (1=1, teste $i = j$)
  - $aabba$ (2=2, teste $i = j$)
  - $abbaa$ (2=2, teste $j = k$)
- **Rejeita:**
  - $aabbb$ (2 3 e 3 0)
  - $aaaba$ (3 2 e 2 1)

Diagrama de Estados

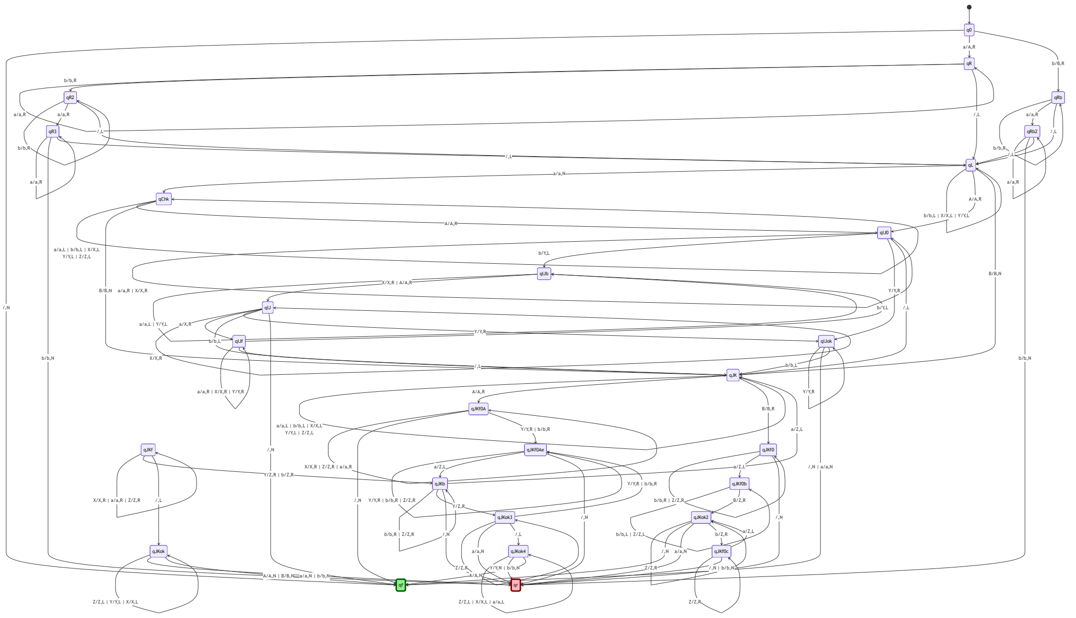

Figura 28: MTND para $L = \{a^i b^j a^k \mid i = j \text{ ou } j = k\}$ com 2 pares não-determinísticos

4.11 Resumo: Comparação de Complexidades (Exercício 4)

Ex	Linguagem	MT Padrão	MTND	Pares ND
4.a	Começa com ab	$O(1)$	$O(1)$	0
4.b	$a^n b^n c^n$	$O(n^2)$	$O(n)$	2
4.c	$a^{2m} b^m$	$O(n^2)$	$O(n)$	2
4.d	Palíndromos ww^R	$O(n^2)$	$O(n)$	2
4.e	$\#a = \#b$	$O(n^2)$	$O(n)$	2
4.f	$1^n 0^{n+3}$	$O(n^2)$	$O(n)$	2
4.g	$a^n b^{2n} c^{n-1}$	$O(n^2)$	$O(n^2)$	0
4.h	$j = \max(i, k)$	$O(n^2)$	$O(n)$	9
4.i	$i = j$ ou $j = k$	$O(n^2)$	$O(n)$	2

Tabela 5: Comparação de complexidades entre MT padrão e MTND

Conclusão

As MTNDs reduzem a complexidade de $O(n^2)$ para $O(n)$ em problemas onde é possível **"adivinhar"uma solução** e verificá-la linearmente:

- **Adivinhar posições:** Onde está o meio de um palíndromo (4.d)
- **Adivinhar pareamentos:** Qual b parear com qual a (4.e)
- **Adivinhar condições:** Qual teste fazer: $i = j$ ou $j = k$ (4.i)
- **Adivinhar quantidades:** Quantos símbolos marcar por iteração (4.b, 4.c)

O não-determinismo permite explorar múltiplas possibilidades em "paralelo", aceitando se **ao menos um caminho** levar à aceitação.

Anexo: Comandos CLI para Testes

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

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

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

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

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