# *5 MÁQUINA DE TURING*

## 5.1 Noção Intuitiva

## 5.2 Noção como Máquina

## 5.3 Modelo Formal

## 5.4 Máquinas de Turing como Reconhecedores de Linguagens

## 5.5 Máquinas de Turing como Processadores de Funções

## 5.6 Conclusões

## 5.7 Exercícios

# *5 MÁQUINA DE TURING*

- proposta por Alan Turing em 1936;
- é universalmente conhecida e aceita como formalização de algoritmo;
- trata-se de um mecanismo simples que formaliza a ideia de uma pessoa que realiza cálculos;
- possui, no mínimo, o mesmo poder computacional de qualquer computador de propósito geral;
- não constitui uma *máquina*, como definida anteriormente, mas sim um programa para uma máquina universal.

## 5.1 Noção Intuitiva

- O ponto de partida de Turing foi analisar a situação na qual uma pessoa, equipada com um instrumento de escrita e um apagador, realiza cálculos em uma folha de papel organizada em quadrados.
- Inicialmente, a folha de papel contém somente os dados iniciais do problema.
- O trabalho da pessoa pode ser resumido em seqüências de operações simples como segue:
  - ler um símbolo de um quadrado;
  - alterar um símbolo em um quadrado;
  - mover os olhos para outro quadrado;
  - quando é encontrada alguma representação satisfatória para a resposta desejada, a pessoa termina seus cálculos.
- Para viabilizar esse procedimento, as seguintes hipóteses são aceitáveis:
  - a natureza bidimensional do papel não é um requerimento essencial para os cálculos.
  - é assumido que o papel consiste de uma fita infinita organizada em quadrados (células);
  - conjunto de símbolos pode ser finito;
  - conjunto de estados da mente da pessoa durante o processo de cálculo é finito.
  - existem dois estados em particular: *estado inicial* e *estado final*, correspondendo ao início e ao fim dos cálculos, respectivamente;
- comportamento da pessoa a cada momento é determinado somente pelo seu estado presente e pelo símbolo para o qual sua atenção está voltada;
- a pessoa é capaz de observar e alterar o símbolo de apenas um quadrado de cada vez, bem como de transferir sua atenção somente para um dos quadrados adjacentes.

# 5.2 Noção como Máquina

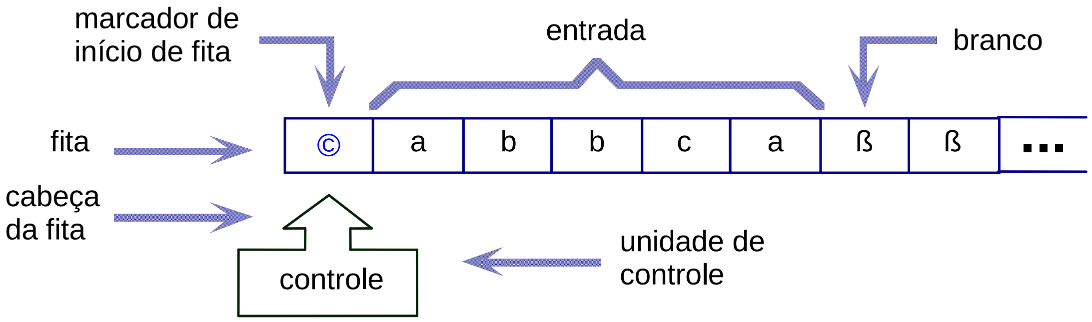



***Figura 5.1 Fita e unidade de controle de uma Máquina de Turing***

## *Fita*.

- Usada simultaneamente como dispositivo de entrada, de saída e de memória de trabalho;
- É finita à esquerda e infinita (tão grande quanto necessário) à direita, sendo dividida em células, cada uma das quais armazenando um símbolo.
- Os símbolos podem pertencer:
  - ⇒ ao alfabeto de entrada,
  - ⇒ ao alfabeto auxiliar
  - ⇒ **ß** branco
  - ⇒ © marcador de início de fita
- Inicialmente, a palavra a ser processada ocupa as células mais à esquerda, após o marcador de início de fita, ficando as demais com *branco*.

## *Unidade de Controle*

- Reflete o estado corrente da máquina.
- Possui um número finito e predefinido de estados.

- Possui uma unidade de leitura e gravação (*cabeça da fita*), a qual acessa uma célula da fita de cada vez.
- A *cabeça da fita* lê o símbolo de uma célula de cada vez e grava um novo símbolo. Após a leitura/gravação (a gravação é realizada na mesma célula de leitura), a cabeça move-se uma célula para a direita ou esquerda.

***Programa*** ou ***Função de Transição***.

- o programa comanda as leituras e gravações, o sentido de movimento da cabeça e define o estado da máquina.
- programa é uma função que, dependendo do estado corrente da máquina e do símbolo lido, determina o símbolo a ser gravado, o sentido do movimento da cabeça e o novo estado.

# 5.3 Modelo Formal

### *Definição 5.1 Máquina de Turing.*

Uma ***Máquina de Turing*** é uma 8-upla:

$$M = (\Sigma, Q, \Pi, q_0, F, V, ß, ©)$$

$\Sigma$ *alfabeto de símbolos de entrada*;
$Q$ *conjunto de estados* possíveis da máquina, o qual é finito;
$\Pi$ *programa* ou *função de transição*: (é uma função parcial)

$$\Pi: Q \times (\Sigma \cup V \cup \{ ß, © \}) \rightarrow Q \times (\Sigma \cup V \cup \{ ß, © \}) \times \{ E, D \}$$

$q_0$ *estado inicial* da máquina, tal que $q_0$ é elemento de $Q$;
$F$ *conjunto de estados finais,* tal que $F$ está contido em $Q$;
$V$ *alfabeto auxiliar* ;
ß símbolo especial *branco*;
© símbolo especial *marcador de início* da fita.

- símbolo de início de fita ocorre exatamente uma vez e sempre na célula mais à esquerda da fita, auxiliando na identificação de que a cabeça da fita se encontra na célula mais à esquerda da fita.

- A função programa considera:
  - estado corrente $p \in Q$,
  - símbolo lido da fita $a_u \in (\Sigma \cup V \cup \{ ß, © \})$

para determinar:

- novo estado $q \in Q$,
- símbolo a ser gravado $a_v \in (\Sigma \cup V \cup \{ ß, © \})$
- sentido de movimento da cabeça esquerda ($E$) e direita ($D$) $m \in \{E, D\}$

O programa pode ser representado como um grafo finito

$\Pi(p, a_u) = (q, a_v, m)$

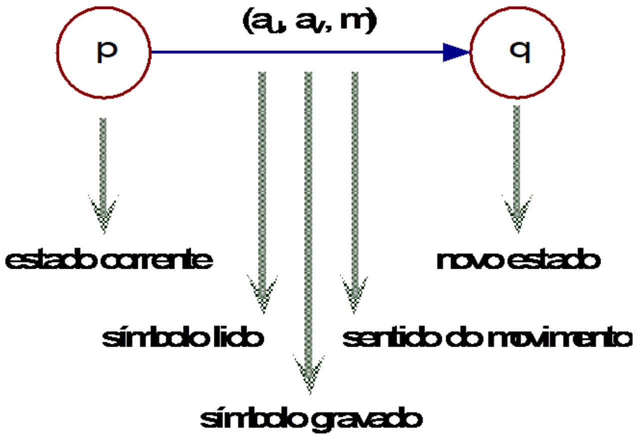



***figura 5.2 Representação da função programa como um grafo***


***figura 5.3 Representação de um estado inicial (esq.) e final (dir.) como nodos de grafos***

O programa pode ser representado por uma Tabela de Transições

$\Pi(p, a_u) = (q, a_v, m)$

| $\Pi$ | © | … | $a_u$ | … | $a_v$ | … | ß |
|---|---|---|---|---|---|---|---|
| p | | | $(q, a_v, m)$ | | | | |
| q | | | | | | | |
| … | | | | | | | |

***figura 5.4 Representação da Função Programa como uma tabela***

- O processamento de uma Máquina de Turing $M=(\Sigma, Q, \Pi, q_0, F, V, ß, ©)$ para uma palavra de entrada w consiste na sucessiva aplicação da função programa, a partir do estado inicial $q_0$ e da cabeça posicionada na célula mais à esquerda da fita até ocorrer uma condição de parada.
- processamento de M para a entrada w pára ou fica em *loop* infinito.

- A **parada** pode ser de duas maneiras: aceitando ou rejeitando a entrada w.

➢ As condições de **parada** são as seguintes:

***Estado Final.***

A máquina assume um estado final: a máquina pára, e a palavra de entrada é **aceita**;

***Função Indefinida.***

A função programa é indefinida para o argumento (símbolo lido e estado corrente): a máquina pára, e a palavra de entrada é **rejeitada**;

***Movimento Inválido.***

O argumento corrente da função programa define um movimento à esquerda e a cabeça da fita já se encontra na célula mais à esquerda: a máquina pára, e a palavra de entrada é **rejeitada**.

***Observação 5.3***

***Definição Alternativa de Máquina de Turing.***

➢ Diversas variações sobre a definição de Máquina de Turing são adotadas.

➢ Note-se que estas variações não alteram o poder computacional do formalismo.

➢ As variações mais significativas estão nas características da fita e no movimento da cabeça como, por exemplo:

***Inexistência do marcador de início de fita.***

- É freqüente não incluir um marcador de início de fita.
- a célula mais à esquerda da fita contém o primeiro símbolo da entrada (ou branco, se a entrada for vazia) .
- Na função programa, deve-se tomar cuidado especial para controlar quando a cabeça da fita atinge o fim da mesma.

***Cabeça de fita não se move em leitura/gravação.***

- Na função programa, é possível especificar que a cabeça permaneça parada, adicionalmente ao movimento para esquerda ou direita.
- O principal objetivo dessa variação é facilitar a especificação da função programa, bem como reduzir o número de transições necessárias.

## 5.4 Máquinas de Turing como reconhecedores de linguagens

- Uma das abordagens do estudo das Máquinas de Turing é o reconhecimento de linguagens,
- Reconhecedores são dispositivos capazes de determinar se uma dada palavra sobre o alfabeto de entrada pertence ou não a uma certa linguagem.

***Definição 5.4 linguagem aceita por uma máquina de turing.***

Seja $M = (\Sigma, Q, \Pi, q_0, F, V, ß, ©)$ uma Máquina de Turing.

**Então:**

**a)** A ***linguagem aceita*** por M, denotada por ACEITA(M) ou L(M), é o conjunto de todas as palavras pertencentes a $\Sigma^*$ aceitas por M,

**ACEITA(M)** $= \{w \mid M$ ao processar $w \in \Sigma^*$, pára em um estado $q_f \in F\}$

**b)** A ***linguagem rejeitada*** por M, denotada por REJEITA(M), é o conjunto de todas as palavras de $\Sigma^*$ rejeitadas por M,

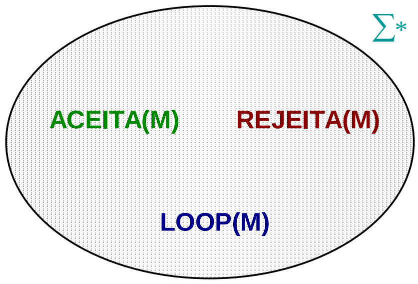
**REJEITA(M)** $= \{w \mid M$ ao processar $w \in \Sigma^*$, pára em um estado $q \notin F\}$

**c)** A ***linguagem para a qual M fica em loop*** infinito,

**LOOP(M)** é conjunto de todas as palavras de $\Sigma^*$ para as quais M fica processando indefinidamente.

- As seguintes afirmações são verdadeiras:

$\text{ACEITA}(M) \cap \text{REJEITA}(M) = \varnothing$
$\text{ACEITA}(M) \cap \text{LOOP}(M) = \varnothing$
$\text{REJEITA}(M) \cap \text{LOOP}(M) = \varnothing$
$\text{ACEITA}(M) \cap \text{REJEITA}(M) \cap \text{LOOP}(M) = \varnothing$
$\text{ACEITA}(M) \cup \text{REJEITA}(M) \cup \text{LOOP}(M) = \Sigma^*$




- O **complemento** de:

  $\text{ACEITA}(M)$ é $\text{REJEITA}(M) \cup \text{LOOP}(M)$
  $\text{REJEITA}(M)$ é $\text{ACEITA}(M) \cup \text{LOOP}(M)$
  $\text{LOOP}(M)$ é $\text{ACEITA}(M) \cup \text{REJEITA}(M)$

### *Exemplo 5.1 Máquina de Turing – duplo balanceamento.*

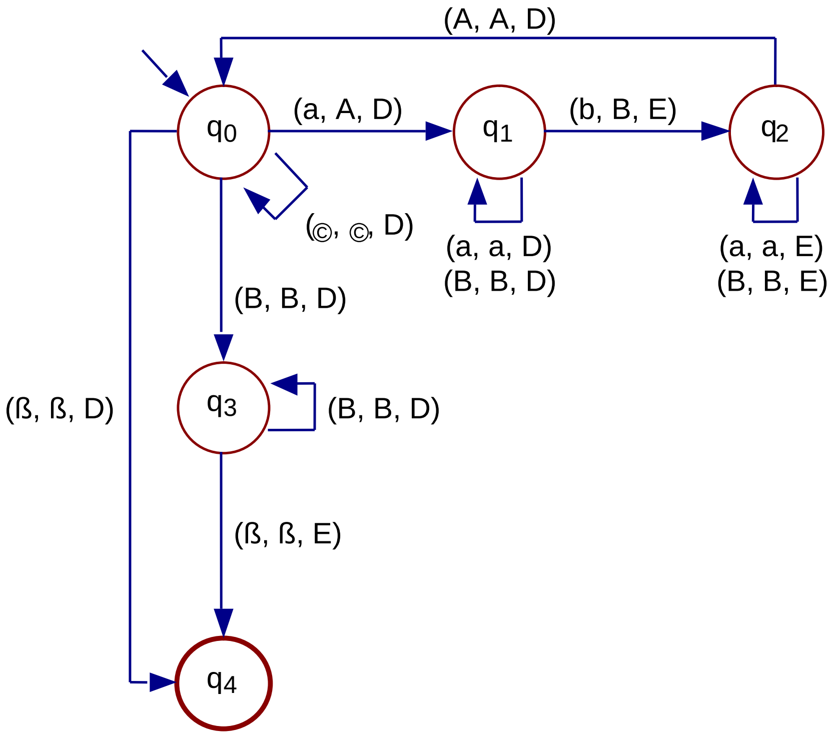
Considere a linguagem:

***Duplo_Bal*** $= \{ a^n b^n \mid n \geq 0 \}$

A Máquina de Turing:

MT_Duplo_Bal=({a,b},{$q_0$, $q_1$, $q_2$, $q_3$, $q_4$}, Π, $q_0$, {$q_4$}, {A,B}, ß, ©)

ACEITA(MT_Duplo_Bal) = Duplo_Bal
REJEITA(MT_Duplo_Bal) = Σ* - Duplo_Bal
LOOP(MT_Duplo_Bal) = ∅




| Π | © | a | b | A | B | ß |
|---|---|---|---|---|---|---|
| $q_0$ | ($q_0$, ©, D) | ($q_1$, A, D) | | | ($q_3$, B, D) | ($q_4$, ß, D) |
| $q_1$ | | ($q_1$, a, D) | ($q_2$, B, E) | | ($q_1$, B, D) | |
| $q_2$ | | ($q_2$, a, E) | | ($q_0$, A, D) | ($q_2$, B, E) | |
| $q_3$ | | | | | ($q_3$, B, D) | ($q_4$, ß, E) |
| $q_4$ | | | | | | |

***Figura 5.6 Grafo e tabela de transições da Máquina de Turing – Duplo_Bal***

## *Descrição do Programa:*

- *programa reconhece o primeiro símbolo a, o qual é marcado como A, e movimenta a cabeça da fita para a direita, procurando o b* ***correspondente, o qual é marcado como B.***
- *Esse ciclo é repetido sucessivamente até identificar, para cada a, o seu correspondente b.*
- *programa garante que qualquer outra palavra que não esteja na forma $a^nb^n$ é rejeitada.*
- *Note que o símbolo de início de fita não tem influência na solução.*

***Computação: máquina de turing duplo_bal para a entrada w = aabb.***

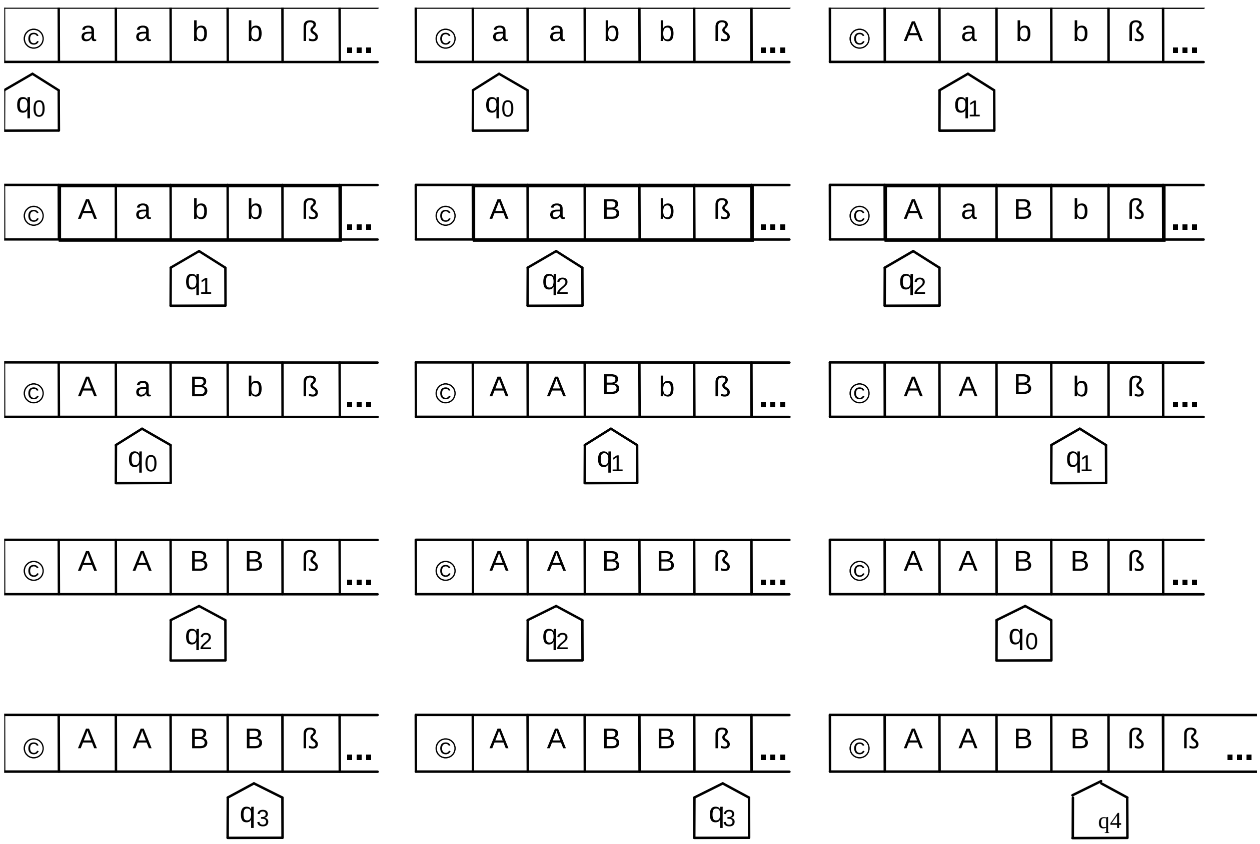



***Figura 5.7 Computação de uma Máquina de Turing***

- Critério para o Reconhecimento de Linguagens.
  *Se a máquina pára para toda palavra da linguagem sobre o alfabeto de entrada, ela é reconhecida pela Máquina de Turing.*

### ***Definição 5.5 Linguagem Enumerável Recursivamente.***

Uma linguagem aceita por uma Máquina de Turing é dita *enumerável recursivamente*.

- ***Enumerável*** deriva do fato de que as palavras de qualquer linguagem *enumerável recursivamente* podem ser enumeradas ou listadas por uma Máquina de Turing.
- ***Recursivamente*** é um termo matemático, anterior ao computador, com significado similar ao de recursão, utilizado na computação.

- A ***Classe das Linguagens Enumeráveis Recursivamente*** inclui as linguagens livre do contexto e algumas outras linguagens para as quais não se pode, mecanicamente, determinar se uma dada palavra pertence ou não à linguagem.
  - Se L é uma dessas linguagens, então para qualquer máquina M que aceita a linguagem L, existe pelo menos uma palavra w, não pertencente a L, que, ao ser processada por M, resulta que a máquina entre em *loop* infinito.
    a) Se w pertence a L, M pára e aceita a entrada;
    b) Se w não pertence a L(w pertence ao complemento de L), M pode parar, rejeitando a palavra, ou permanecer processando indefinidamente (*loop*).

### ***Exemplo 5.2 Linguagem Enumerável Recursivamente.***

As seguintes linguagens são exemplos de linguagens Enumeráveis Recursivamente.

a) Duplo_Bal = $\{ a^n b^n \,/\, n \geq 0\}$
b) Triplo_Bal = $\{ a^n b^n c^n \,/\, n \geq 0\}$
c) Palavra_Palavra = { ww / w é palavra sobre os símbolos a e b}
d) { w / w tem o mesmo número de símbolos a que b}
e) $\{ a^i b^j c^k \,/\, i=j \text{ ou } j=k\}$

Uma sub-classe da ***Classe das Linguagens Enumerável Recursivamente***, denominada ***Classe das Linguagens Recursivas***, é composta pelas linguagens para as quais existe pelo menos uma Máquina de Turing que pára para qualquer entrada, aceitando ou rejeitando.

***Definição 5.6 Linguagem Recursiva.***

Uma linguagem é dita *recursiva* se existe uma Máquina de Turing tal que:

**ACEITA(M) = L**

**REJEITA(M) = $\Sigma^*$ - L.**

**LOOP(M) = $\varnothing$**

- Pode-se afirmar que a Classe das Linguagens Recursivas representa todas as linguagens que podem ser reconhecidas mecanicamente.
  - Existem conjuntos que não são Enumeráveis Recursivamente, ou seja, linguagens para as quais não é possível desenvolver uma Máquina de Turing que as reconheça.
  - O cardinal do conjunto dessas linguagens que não são Enumeráveis Recursivamente é **não contável** (muito grande).

*EXEMPLO 5.3 Linguagem Recursiva*

São exemplos de linguagens recursivas:

a) Duplo_Bal = { $a^n b^n$ / $n \geq 0$}

b) Triplo_Bal ={ $a^n b^n c^n$ / $n \geq 0$}

c) { w / w$\in$\{a,b\}* tem o dobro de símbolos a que b}

**Propriedades das linguagens recursivas:**

**a) Se uma linguagem L sobre um alfabeto $\Sigma$ qualquer é recursiva, então seu complemento, ou seja, $\Sigma^*$-L, é recursivo.**

**b) Uma linguagem L sobre um alfabeto $\Sigma$ qualquer é recursiva se, e somente se, L e seu complemento são enumeráveis recursivamente.**

**c) A Classe das Linguagens Recursivas está contida propriamente na Classe das Linguagens Enumeráveis Recursivamente.**

## a) Complemento de uma Linguagem Recursiva é uma Linguagem Recursiva.

Se uma linguagem L sobre um alfabeto Σ qualquer é recursiva, então o seu complemento Σ* - L também é uma linguagem recursiva.

**PROVA**

- Suponha L uma linguagem recursiva sobre Σ.
  - Então existe M, Máquina Universal, que aceita a linguagem e sempre pára para qualquer entrada. Ou seja:

    ACEITA(M) = L

    REJEITA(M) = Σ* - L

    LOOP(M) = ∅
- Seja M' uma Máquina Universal construída a partir de M, mas invertendo-se as condições de ACEITA por REJEITA e vice-versa.
- Portanto, M' aceita Σ* - L e sempre pára para qualquer entrada. Ou seja:

  ACEITA(M') = Σ* - L

  REJEITA(M') = L

  LOOP(M') = ∅

⇒ **Logo Σ* - L é uma linguagem recursiva.**

## b) *Linguagem Recursiva × Linguagem Enumerável Recursivamente.*

Uma linguagem L sobre um alfabeto Σ qualquer é recursiva se, e somente se, L e Σ* - L são enumeráveis recursivamente.

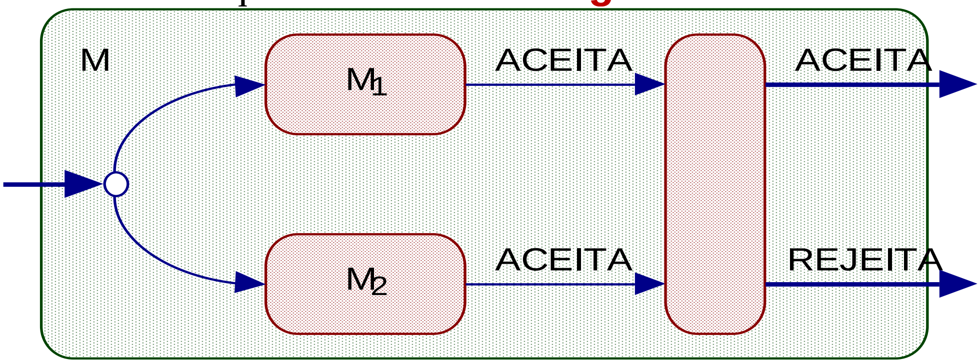
**PROVA**

**a) Suponha L uma linguagem recursiva sobre Σ.**

- Então, como foi mostrado no item anterior, Σ* - L é recursiva.
- Como toda linguagem recursiva também é enumerável recursivamente,
  ⇒ **então L e Σ* - L são enumeráveis recursivamente;**

**b) Suponha L uma linguagem sobre Σ tal que L e Σ* - L são enumeráveis recursivamente.**

- Então existem $M_1$ e $M_2$, Máquinas Universais tais que:
  ACEITA($M_1$) = L
  ACEITA($M_2$) = Σ* - L
- Seja M ***Máquina Universal não-determinística*** definida conforme esquema ilustrado na **figura**.



- P ara qualquer palavra de entrada, M aceita-a se $M_1$ aceitá-la e M rejeita-a se $M_2$ aceitá-la.

  $\Rightarrow$ **Portanto, claramente, M sempre pára.**
  $\Rightarrow$ **Logo, L é recursiva.**

# 5.5 Máquinas de Turing como Processadores de Funções

O estudo é restrito às funções de mapeamento de palavras de um alfabeto $\Sigma$ em uma palavra do mesmo alfabeto.

### *Definição 5.7 Função Turing-Computável.*

Uma função parcial:

$$f: (\Sigma^*)^n \to \Sigma^*$$

é dita *Função Turing-Computável* ou simplesmente *Função Computável* se existe uma Máquina de Turing:

$$M = (\Sigma, Q, \Pi, q_0, F, V, ß, ©)$$

que computa f, ou seja:

**a)** Para $(w_1, w_2, \ldots, w_n) \in (\Sigma^*)^n$, tem-se que a palavra de entrada para M é: © $w_1\ w_2 \ldots w_n$

**b)** Se f é definida para $(w_1, w_2, \ldots, w_n)$, então o processamento de M para a entrada © $w1\ w2 \ldots wn$ é tal que: © w

- pára (aceitando ou rejeitando);
- o conteúdo da fita é (excetuando-se os símbolos brancos):

**c)** Se f é indefinida para $(w_1, w_2, \ldots, w_n)$, então M, ao processar a entrada © $w_1\ w_2 \ldots w_n$ , fica em *loop* infinito.

**Observação 5.8: Função Turing-Computável × Condição de Parada.**

- Na definição acima, é considerado como resultado do processamento de M somente o conteúdo gravado na fita, sendo irrelevante o estado de parada da máquina.
- Portanto, relativamente a função computável, um processamento que pára em um estado não-final tem seu resultado definido.
- Se a função computável não for definida para a entrada, é porque o processamento é infinito.

### *Definição 5.10 Função Turing-Computável Total.*

Uma função total: $f: (\Sigma^*)^n \to \Sigma^*$

*é dita Função Turing-Computável Total ou simplesmente Função Computável Total se existe uma Máquina de Turing $M = (\Sigma, Q, \Pi, q0, F, V, ß, ©)$ que computa f e sempre pára para qualquer entrada.*

## *Exemplo 5.4 máquina de Turing – concatenação.*

A função **concatenação**: $(\{ a, b \}^*)^n \rightarrow \{ a, b \}^*$

é tal que associa ao par $(w_1, w_2)$ a palavra $w_1w_2$.

A Máquina de Turing:

**Conc = ({ a, b, # }, { $q_0$, $q_1$, $q_2$, $q_3$, $q_4$ }, Π, $q_0$, { $q_4$ }, ∅, ß, ©)**

| Π | © | a | b | # | ß |
|---|---|---|---|---|---|
| $q_0$ | ($q_0$, ©, D) | ($q_0$, a, D) | ($q_0$, b, D) | ($q_0$, #, D) | ($q_1$, ß, E) |
| $q_1$ | | ($q_2$, ß, E) | ($q_3$, ß, E) | ($q_4$, ß, E) | |
| $q_2$ | | ($q_2$, a, E) | ($q_3$, a, E) | ($q_4$, a, E) | |
| $q_3$ | | ($q_2$, b, E) | ($q_3$, b, E) | ($q_4$, b, E) | |
| $q_4$ | | | | | |

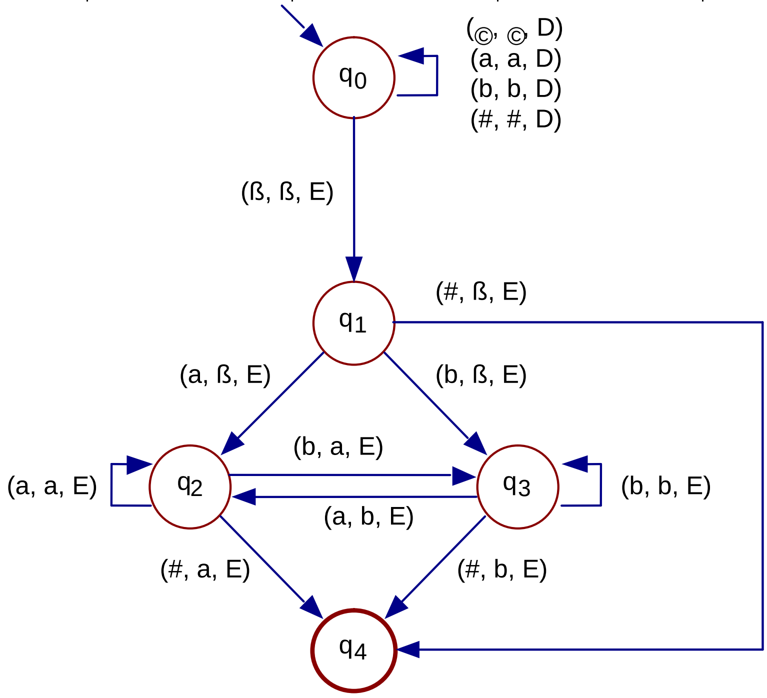



***figura 5.8 Tabela de Transições e Grafo da Máquina de Turing Concatenação***

## *Descrição do Programa:*

- O programa recebe como entrada a palavra: © $w_1$ # $w_2$.
- posiciona a cabeça no último símbolo da palavra de entrada.
- move a cabeça para a esquerda até encontrar o símbolo #, quando pára. Enquanto move a cabeça, ao ler um símbolo, grava sobre este o símbolo lido anteriormente.
- A memorização do símbolo anterior é realizada pelos estados como segue:

- q2
  memoriza que o símbolo anterior é a;
- q3
  memoriza que o símbolo anterior é b.

## *Exemplo 5.5 máquina de Turing – função quadrado.*

- A função **quadrado**: { 1 }* → { 1 }* é tal que associa o valor natural n, representado em unário, ao valor $n^2$ (também em unário).
- A Máquina de Turing:

**Quadr = ({1}, { $q_0$, $q_1$, $q_2$, ..., $q_{13}$ }, Π, $q_0$, { $q_{13}$ }, { A, B, C}, ß, ©)**

| Π | © | 1 | A | B | C | ß |
|---|---|---|---|---|---|---|
| $q_0$ | ($q_0$, ©, D) | ($q_1$, A, D) | | ($q_0$, B, D) | | ($q_3$, ß, E |
| $q_1$ | | ($q_1$, 1, D) | | ($q_1$, B, D) | | ($q_2$, B, E |
| $q_2$ | | ($q_2$, 1, E) | ($q_0$, A, D) | ($q_2$, B, E) | | |
| $q_3$ | ($q_{13}$, ©, D) | | | ($q_4$, ß, E) | | |
| $q_4$ | | | ($q_5$, A, D) | ($q_4$, B, E) | | |
| $q_5$ | | | | ($q_6$, C, E) | | ($q_{12}$, ß, E |
| $q_6$ | ($q_7$, ©, D) | | ($q_6$, 1, E) | | ($q_6$, C, E) | |
| $q_7$ | | ($q_8$, A, D) | | | | |
| $q_8$ | | ($q_9$, A, D) | | | ($q_{11}$, C, D) | |
| $q_9$ | | ($q_9$, 1, D) | | ($q_9$, B, D) | ($q_9$, C, D) | ($q_{10}$, 1, E |
| $q_{10}$ | | ($q_{10}$, 1, E) | ($q_8$, A, D) | ($q_{10}$, B, E) | ($q_{10}$, C, E) | |
| $q_{11}$ | | ($q_{12}$, 1, E) | | ($q_6$, C, E) | ($q_{11}$, C, D) | |
| $q_{12}$ | ($q_{13}$, ©, D) | | ($q_{12}$, 1, E) | | ($q_{12}$, 1, E) | |
| $q_{13}$ | | | | | | |

- O programa recebe como entrada a palavra: © $n_1$ , onde $n_1$ denota o valor n representado em unário sobre { 1 }.
- $(n_1)^2$ é simplesmente $n_1$ concatenado consigo mesmo, ou seja:

$$(n_1)^2 = n_1 \bullet n_1$$

***Descrição do Programa:*** A concatenação é obtida por:

- Em $q_0$, $q_1$ e $q_2$, é gerado ©$n_A$ $n_B$ ($n_A$ e $n_B$ são em unário sobre {A} e {B});
- Em $q_0$, $q_3$ e $q_4$, é retirado um símbolo B de $n_B$, resultando em $n_A$ $(n-1)_B$
- Em $q_5$ até $q_{11}$, a subpalavra $(n-1)_B$ é usada para controlar concatenações sucessivas, resultando em © $n_A$ $(n-1)_C$ $(n-1)_1$ $(n-1)_1$ ... $(n-1)_1$, onde $(n-1)_1$ é repetida n-1 vezes
- Em $q_{12}$, as subpalavras $n_A(n-1)_C$ são substituídas por $n_1(n-1)_1$, resultando em ©$n_1$ $(n-1)_1$ $(n-1)_1$ $(n-1)_1$ ... $(n-1)_1$ onde $(n-1)_1$ é repetida n vezes
- ou seja, o comprimento da palavra resultante é:

$$n + (n-1) \bullet n = n + (n^2 - n) = n^2$$

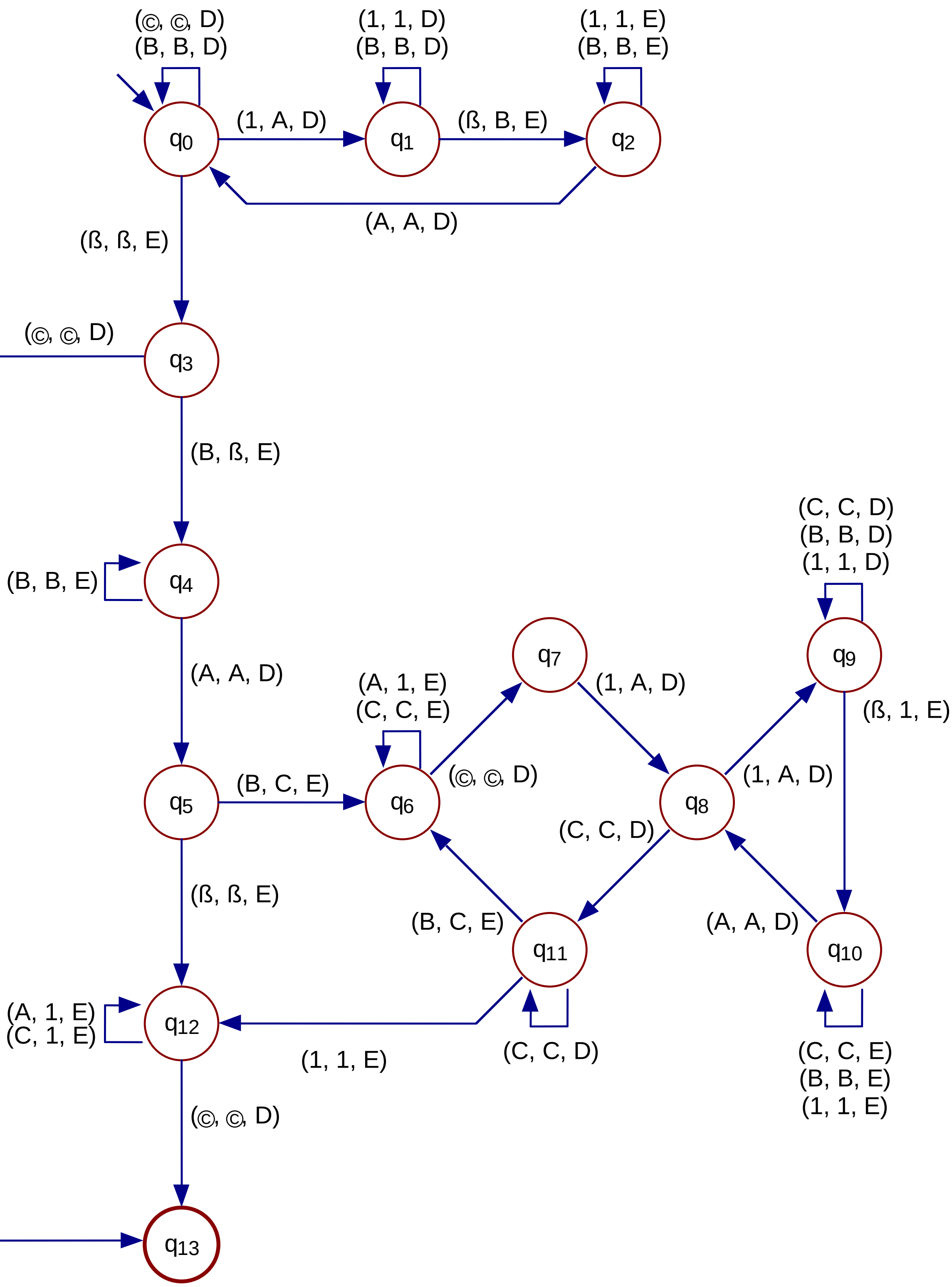



***figura 5.10 grafo da máquina de turing – função quadrado***

***Computação: máquina de Turing quadrado para a entrada w = 111.***

| | | |
|---|---|---|
| (q0)©111® | © (q6)A11CB ® | ©AA1CC(q9)11 ® |
| ©(q0)111 ® | (q6)©111CB ® | ©AA1CC1(q9)1 ® |
| ©A(q1)11 ® | © (q7)111CB ® | ©AA1CC11(q9) ® |
| ©A1(q1)1 ® | ©A(q8)11CB ® | ©AA1CC1(q10)11 ® |
| ©A11(q1) ® | ©AA(q9)1CB ® | ©AA1CC(q10)111 ® |
| ©A1(q2)1B ® | ©AA1(q9)CB ® | ©AA1C(q10)C111 ® |
| ©A(q2)11B ® | ©AA1C(q9)B ® | ©AA1(q10)CC111 ® |
| © (q2)A11B ® | ©AA1CB(q9) ® | ©AA(q10)1CC111 ® |
| ©A(q0)11B ® | ©AA1C(q10)B1 ® | ©A(q10)A1CC111 ® |
| ©AA(q1)1B ® | ©AA1(q10)CB1 ® | ©AA(q8)1CC111 ® |
| ©AA1(q1)B ® | ©AA(q10)1CB1 ® | ©AAA(q9)CC111 ® |
| ©AA1B(q1) ® | ©A(q10)A1CB1 ® | ©AAAC(q9)C111 ® |
| ©AA1(q2)BB ® | ©AA(q8)1CB1 ® | ©AAACC(q9)111 ® |
| ©AA(q2)1BB ® | ©AAA(q9)CB1 ® | ©AAACC1(q9)11 ® |
| ©A(q2)A1BB ® | ©AAAC(q9)B1 ® | ©AAACC11(q9)1 ® |
| ©AA(q0)1BB ® | ©AAACB(q9)1 ® | ©AAACC111(q9) ® |
| ©AAA(q1)BB ® | ©AAACB1(q9) ® | ©AAACC11(q10)11 ® |
| ©AAAB(q1)B ® | ©AAACB(q10)11 ® | ©AAACC1(q10)111 ® |
| ©AAABB(q1) ® | ©AAAC(q10)B11 ® | ©AAACC(q10)1111 ® |
| ©AAAB(q2)BB ® | ©AAA(q10)CB11 ® | ©AAAC(q10)C1111 ® |
| ©AAA(q2)BBB ® | ©AA(q10)ACB11 ® | ©AAA(q10)CC1111 ® |
| ©AA(q2)ABBB ® | ©AAA(q8)CB11 ® | ©AA(q10)ACC1111 ® |
| ©AAA(q0)BBB ® | ©AAAC(q11)B11 ® | ©AAA(q8)CC1111 ® |
| ©AAAB(q0)BB ® | ©AAA(q6)CC11 ® | ©AAAC(q11)C1111 ® |
| ©AAABB(q0)B ® | ©AA(q6)ACC11 ® | ©AAACC(q11)1111 ® |
| ©AAABBB(q0) ® | ©A(q6)A1CC11 ® | ©AAAC(q12)C1111 ® |
| ©AAABB(q3)B ® | ©(q6)A11CC11 ® | ©AAA(q12)C11111 ® |
| ©AAAB(q4)B ® | (q6)©111CC11 ® | ©AA(q12)A111111 ® |
| ©AAA(q4)BB ® | ©(q7)111CC11 ® | ©A(q12)A1111111 ® |
| ©AA(q4)ABB ® | ©A(q8)11CC11 ® | ©(q12)A11111111 ® |
| ©AAA(q5)BB ® | ©AA(q9)1CC11 ® | (q12)©111111111 ® |
| ©AA(q6)ACB ® | ©AA1(q9)CC11 ® | ©(q13)111111111 ® |
| ©A(q6)A1CB ® | ©AA1C(q9)C11 ® | |

***figura 5.11 computação da máq de turing função quadrado***

## 5.6 Conclusões

Foi visto a Máquina de Turing como um dos modelos mais utilizado para a formalização de algoritmo. Trata-se de um formalismo simples, poderoso e que formaliza a idéia de uma pessoa que realiza cálculos. É muito similar a um computador moderno, embora tenha sido proposto muitos anos antes do primeiro computador digital.

Duas das abordagens da Máquina de Turing foram apresentadas: reconhecimento de linguagens e processamento de funções.

Uma terceira abordagem é a solucionabilidade de problemas, a qual será abordada em capítulo subsequente.

Algumas evidências de que se trata de uma máquina universal foram apresentadas.

## 5.7 Exercícios

### Exercício 5.1

Qual a importância do estudo da Máquina de Turing na Ciência da Computação?

A Máquina de Turing na Ciência da Computação é de grande importância, pois formaliza a noção intuitiva de algoritmo. Também formaliza o conceito de uma máquina universal, ou seja, uma máquina na qual qualquer programa computável (que tem uma máquina de turing que o resolva) roda.

### Exercício 5.2

Desenvolva Máquinas de Turing, que aceitem as seguintes linguagens:

a) $L_1 = \varnothing$

b) $L_2 = \{ \varepsilon \}$

c) $L_3 = \{ w \mid w$ tem o mesmo número de símbolos $a$ e $b \}$

d) $L_4 = \{ w \mid$ o décimo símbolo da direita para a esquerda é $a \}$

e) $L_5 = \{ waw \mid w$ é palavra de $\{ a, b \}^* \}$

f) $L_6 = \{ ww \mid w$ é palavra de $\{ a, b \}^* \}$

g) $L_7 =\{ww^r \mid w$ é palavra de $\{a,b\}^*$ / $w^r$ denota a palavra reversa de $w\}$

h) $L_8 = \{ www \mid w$ é palavra de $\{ a, b \}^* \}$

i) $L_9 = \{ w \mid w = a^1 b^2 a^3 b^4 \ldots a^{n-1} b^n$ e $n$ é número natural par $\}$

j) $L_{10} = \{ w \mid w = a^n b^n$ ou $w = b^n a^n \}$

k) $L_{11} = \{ w \mid w = a^i b^j c^k$, onde ou $i = j$ ou $j = k \}$

l) $Triplo\_Bal = \{ a^n b^n c^n \mid n \geq 0 \}$

m) $Dobro = \{ w \mid w \in \{ a, b \}^*$ e tem o dobro de símbolos $a$ que $b \}$

**Exercício 5.3**

Desenvolva Máquinas de Turing, que realizam as operações e testes abaixo:

a) A – B: subtração nos inteiros
b) div(A, B): divisão inteira de A por B
c) fat(A): fatorial;
d) pot(A, B): potência;
e) teste(A > B): nos inteiros
f) teste(A ≥ B): nos inteiros
g) teste(A ≤ B): nos inteiros
h) mdc(A, B): máximo divisor comum;
i) teste_primo_entre_si(A, B): verifica se são primos entre si, mdc(A,B)=1;
j) mod(A, B): resto da divisão inteira;
k) teste_primo(A): verifica se A é primo;
l) teste_nperf(A): verifica se é um número perfeito.

**Exercício 5.4**

Seja a expressão booleana (EB) definida indutivamente como segue:

i) v e f são EB;
ii) Se p e q são EB, então p e q e p ou q também são EB.

Assuma que o conetivo e tem prioridade sobre o ou. Então:
Construa uma Máquina de Turing sobre Σ = { v, f, e, ou } tal que:

ACEITA(M) = { EB | EB é v }
REJEITA(M) = { EB | EB é f }
LOOP(M) = { w | w não é EB }

**Exercício 5.5**

Desenvolva uma Máquina de Turing que receba com entrada uma palavra sobre o alfabeto {a, b, c} sem qualquer ordem e ordene os símbolos supondo que a < b e b < c. Por exemplo, para a entrada abccba, o resultado é aabbcc.

**Exercício 5.6**

Desenvolva uma Máquina de Turing sobre o alfabeto { (, ) } que verifique o correto uso de parênteses em expressões matemáticas. Por exemplo, são expressões válidas:

( )
((( )))
((( ) ( ) ) ( ) )

**Exercício 5.7**

Desenvolva Máquinas de Turing, que aceitem as seguintes linguagens:

*Sugestão*: procure usar Máquinas de Turing já definidas em exercícios anteriores e implemente uma noção de macro ou de composição de máquinas de forma modular.

a) $L_{12} = \{ (ww^r)^3 \mid w$ é palavra de $\{ a,b \}^*$ e $w^r$ denota a palavra reversa de $w \}$

b) $L_{13} = \{ ww^rw^rw \mid w$ é palavra de $\{ a,b \}^*$ e $w^r$ denota a palavra reversa de $w \}$

c) $L_{14} = \{ (a^nb^n)^m \mid n$ e $m$ são números naturais $\}$

**Exercício 5.8**

Qual a diferença fundamental entre as Classes das Linguagens Recursivas e a das Linguagens Enumeráveis Recursivamente? Qual a importância de se distinguir essas duas classes?

- Linguagem Enumerável Recursiva: É uma linguagem na qual as palavras não podem ser determinadas mecanicamente como pertencentes ou não à linguagem. Do conjunto de palavras pertencentes à linguagem a máquina pára. As palavras não pertencentes à linguagem podem tanto fazer a máquina parar, rejeitando-as ou ficar em processamento indefinidamente, visto que as linguagens enumeráveis recursivamente podem possuir loop diferente de vazio.
- Linguagens Recursivas: É uma linguagem que contém as mesmas características da linguagem enumerável recursiva, com a diferença de que a linguagem recursiva não possui loop, ou seja, o seu loop é sempre vazio. Desta forma podemos facilmente compreender que a linguagem enumerável recursiva faz parte do conjunto de linguagens recursivas.

**Exercício 5.9**

Demonstre que a Classe das Linguagens Recursivas é fechada para as operações de união, intersecção e diferença. Demonstre inicialmente para a operação sobre duas linguagens recursivas. Após, amplie a demonstração para $n$ linguagens recursivas (sugestão: demonstre por indução em $n$).

**Exercício 5.10**

Dê a Máquina de Turing MT, sobre o alfabeto $\{ a, b \}$, tal que:

ACEITA(MT) = $\{ w \mid w$ tem o mesmo número de símbolos $a$ e $b \}$

REJEITA(MT)= { w | w cuja diferença entre o número de símbolos a e b é 1 }

LOOP(MT)= { a, b }* - (ACEITA(MT) ∪ REJEITA(MT))

Por exemplo:

ab ∈ ACEITA(MT)
aba ∈ REJEITA(MT)
aaba ∈ LOOP(MT).

**Exercício 5.11**

Considere a seguinte Máquina de Turing cuja função programa Π é como na **figura.12**:

M = ({ a, b }, { q0, q1, q2, q3, q4, q5, q6, q7 }, Π, q0, { q7 }, { A, B }, β, ©)

| Π | © | a | b | A | B | β |
|---|---|---|---|---|---|---|
| q0 | (q0, ©, D) | (q1, A, D) | (q0, b, D) | (q0, A, D) | (q0, B, D) | (q4, β, E) |
| q1 | | (q1, a, D) | (q1, b, D) | | (q2, B, E) | (q2, β, E) |
| q2 | (q5, ©, D) | (q2, a, E) | (q3, B, E) | (q2, A, E) | (q2, B, E) | |
| q3 | (q0, ©, D) | (q3, a, E) | (q3, b, E) | (q0, A, D) | | |
| q4 | (q7, ©, D) | | (q6, b, D) | (q4, A, E) | (q4, B, E) | |
| q5 | | (q6, a, D) | | (q5, A, D) | (q5, B, D) | |
| q6 | (q6, ©, D) | (q6, a, D) | (q6, b, D) | (q6, A, D) | (q6, B, D) | (q6, β, E) |
| q7 | | | | | | |

***figura 5.12 Tabela de transições da Máquina de Turing***

a) Verifique qual a configuração final após a computação para as seguintes palavras: ab (aceita), aba (rejeita) e aaba (loop)

b) Qual a linguagem aceita?

c) A linguagem aceita é apenas enumerável recursivamente ou é também recursiva? Por quê?

**Exercício 5.12**

Elabore uma Máquina de Turing MT_Palíndroma (determinística ou não) que sempre pára para qualquer entrada e que reconhece todas as palíndromas (palavras que possuem a mesma leitura da esquerda para a direita e vice-versa) sobre o alfabeto { a, b }. Por exemplo:

aba, abba, babab ∈ ACEITA(MT_Palíndroma)

abab ∈ REJEITA(MT_Palíndroma)

**Exercício 5.13**

Elabore uma Máquina de Turing M sobre o alfabeto { a, b } tal que:

ACEITA(M) = { w | w inicia com a e, após cada a, existe pelo menos um b }

LOOP(M) = { w | w ∉ ACEITA(M) e existe pelo menos um b entre dois a }

REJEITA(M)= { a, b }* - (ACEITA(M) ∪ LOOP(M))

Por exemplo:

ab, abbab ∈ ACEITA(M)

b, baa, baab ∈ REJEITA(M)

aba, baba, abbaba ∈ LOOP(M)

## Exercício 5.14

Verifique se as respectivas funções computáveis são totais:

a) Função concatenação do **exemplo 5.4** Máquina de Turing Concatenação.;
b) Função quadrado do **exemplo 5.5** Máquina de Turing função quadrado.

## Exercício 5.15

Sobre a Máquina de Turing, analise as seguintes afirmações:

I. Uma linguagem aceita por uma máquina de Turing pode ser dita Linguagem Recursiva;
II. A classe das Linguagens Enumeráveis Recursivamente está contida propriamente na classe das Linguagens Recursivas;
III. O complemento de uma Linguagem Recursiva é uma Linguagem Recursiva.

Marque a alternativa correta:

a) Apenas I e II estão corretas;
b) Apenas II está correta;
c) Apenas II e III estão corretas;
d) Apenas I e III estão corretas;
e) I, II e III estão corretas.

## Exercício 5.16

Considere a seguinte Máquina de Turing: $M = (\Sigma, Q, \Pi, q_0, F, V, \beta, ©)$

Marque a alternativa *errada*:

a) ACEITA(M), é o conjunto de todas as palavras pertencentes a $\Sigma^*$ aceitas por M;
b) REJEITA(M), é o conjunto de todas as palavras pertencentes a $\Sigma^*$ rejeitadas por M;
c) $ACEITA(M) \cap REJEITA(M) = \varnothing$
d) $ACEITA(M) \cup REJEITA(M) = \Sigma^*$
e) O complemento de ACEITA(M) é $REJEITA(M) \cup LOOP(M)$

### Exercício 5.17

Considere a seguinte Máquina de Turing cuja função programa Π é como na **figura 5.13**:

M = ({ a, b }, { q0, q1, q2, q3, q4, qf }, Π, q0, { qf }, ∅, β, ©)

Relacione a primeira coluna de acordo com a segunda, considerando o reconhecimento das palavras por M:

| | |
|---|---|
| (1) ∈ ACEITA(M) | (   ) aababa |
| (2) ∈ REJEITA(M) | (   ) abba |
| (3) ∈ LOOP(M) | (   ) bbab |
| | (   ) aabbba |
| | (   ) aaaabba |

Marque a alternativa que corresponde a correta correlação:

a) 1 – 1 – 3 – 2 – 1
b) 3 – 1 – 2 – 2 – 1
c) 2 – 3 – 1 – 1 – 2
d) 3 – 1 – 2 – 1 – 2
e) 1 – 2 – 1 – 3 – 3

| Π | © | a | b | β |
|---|---|---|---|---|
| q0 | (q0, ©, D) | (q0, a, D) | (q1, b, D) | (q4, β, E) |
| q1 | | (q0, a, E) | (q2, b, D) | |
| q2 | | (q3, b, D) | | |
| q3 | | | | (qf, β, E) |
| q4 | | (q2, a, D) | (q3, a, E) | (q4, β, E) |
| qf | | | | |

***figura 5.13 Tabela de transições da Máquina de Turing***

**Exercício 5.18**

Considere a seguinte Máquina de Turing cuja função programa Π é como na **figura 5.14**

({a, b, #}, {q0, q1, q2, q3, q4, q5, q6, q7, q8, qf}, Π, q0, {q8}, {A,B}, β, ©)

- Marque a alternativa correta:

a) ACEITA(M) = { w # $w^r$ | w ∈ { a, b, # }* e $w^r$ é a reversa de w }
b) ACEITA(M) = { w # w | w ∈ { a, b, # }$^+$ }
c) ACEITA(M) = { w # w | w ∈ { a, b, # }* }
d) ACEITA(M) = ∅
e) Nenhuma das alternativas acima esta correta.

| Π | © | a | b | # | A | B | β |
|---|---|---|---|---|---|---|---|
| q0 | (q0, ©, D) | (q1, A, D) | (q5, B, D) | (q7, #, D) | | | (qf, β, |
| q1 | | (q1, a, D) | (q1, b, D) | (q2, #, D) | | | |
| q2 | | (q3, A, E) | | | (q2, A, D) | (q2, B, D) | |
| q3 | | | | (q4, #, E) | (q3, A, E) | (q3, B, E) | |
| q4 | (q8, ©, D) | (q4, a, E) | (q4, b, E) | | (q0, A, D) | (q0, B, D) | |
| q5 | | (q5, a, D) | (q5, b, D) | (q6, #, D) | | | |
| q6 | | | (q3, B, E) | | (q6, A, D) | (q6, B, D) | |
| q7 | | | | | (q7, A, D) | (q7, B, D) | (qf, β, |
| q8 | | | | | | | |
| qf | | | | | | | |

***figura 5.14 Tabela de transições da Máquina de Turing***

**Exercício 5.19**

Sobre a Máquina de Turing, analise as seguintes afirmações:

*I. O termo função Turing-computável reflete a classe das linguagens recursivas;*
*II. A partir da definição de função Turing-computável, um processamento que pára em um estado não-final, não tem resultado definido;*
*III. O conjunto formado pela união de todas as funções Turing-computável totais possui a propriedade de que o conjunto das palavras que deixam a máquina em* loop *é vazio.*

Marque a alternativa correta:

a) Apenas I está correta;
b) Apenas II está correta;
c) Apenas III está correta;
d) Apenas I e II estão corretas;
e) Apenas II e III estão corretas.

**Exercício 5.20**

Sobre a Máquina de Turing, marque a alternativa *errada*:

a) Uma função parcial $f: (\Sigma^*)^n \to \Sigma^*$ é dita Função Turing-Computável ou simplesmente Função Computável se existe uma Máquina de Turing $M = (\Sigma, Q, \Pi, q_0, F, V, \beta, ©)$ que computa $f$;

b) Uma função total $f: (\Sigma^*)^n \to \Sigma^*$ é dita Função Turing-Computável Total ou simplesmente Função Computável Total se existe uma Máquina de Turing $M = (\Sigma, Q, \Pi, q_0, F, V, \beta, ©)$ que computa $f$ e sempre pára para qualquer entrada;

c) A função programa total sempre induz uma função Turing-Computável total;

d) A definição de uma função Turing-Computável Total garante que a função está definida para todos os valores de entrada;

e) Toda Máquina de Turing vista como um reconhecedor de linguagem também pode ser vista como um processador de função.

**Exercício 5.21**

Como, sem perda de generalidade, pode-se supor que a função programa de uma Máquina de Turing seja total?

**Exercício 5.22**

Desenvolva um simulador universal de Máquina de Turing. Recebe como entrada uma representação de uma Máquina de Turing no formato de uma 8-upla ordenada e a palavra de entrada a ser processada. Para um processamento finito, gera como saída: o estado atingido (incluindo a condição aceita/rejeita); o conteúdo da fita; e o número de movimentos realizados pela cabeça de fita.

Opcionalmente, sugere-se a implementação da opção execução passo a passo da máquina simulada, para permitir uma análise da computação.