7 OUTROS MODELOS DE MÁQUINAS UNIVERSAIS

7.1 Máquina de Post

7.2 Máquina com Pilhas

7.3 Autômato com Duas Pilhas

7.4 Máquinas Não-Deterministicas

7.5 Hierarquia de Classes de Máquinas

7 OUTROS MODELOS DE MÁQUINAS UNIVERSAIS

Formalismos equivalentes à Máquina de Turing (note o tipo de estrutura de dados de cada modelo):

a) *Máquina de Post*. Baseada na estrutura de dados do tipo *fila* (o primeiro dado armazenado é o primeiro a ser recuperado);

b) *Máquinas com Pilhas*. Baseada na estrutura de dados do tipo *pilha* (o último dado armazenado é o primeiro a ser recuperado), onde são necessárias pelo menos duas pilhas para simular o mesmo poder computacional de uma fita ou fila. Trata-se de uma família de máquinas com poder computacional variado.

c) *Autômatos com Duas Pilhas*. Analogamente, é baseada na estrutura de dados do tipo pilha, com exatamente duas pilhas, com programas expressos na forma de grafos.

d) *Não-Determinismo*. Máquinas não-deterministicas, em geral, não tem maior poder computacional do que suas versões determinísticas, excetuando a versao de Maquina com uma pilha ou o Automato com uma pilha.

7.1 Máquina de Post.

- A principal característica da Máquina de Post é que usa uma estrutura de dados do tipo *fila* para entrada, saída e memória de trabalho.
- E struturalmente, a principal característica de uma fila é que o primeiro valor gravado é também o primeiro a ser lido (uma leitura exclui o dado lido).

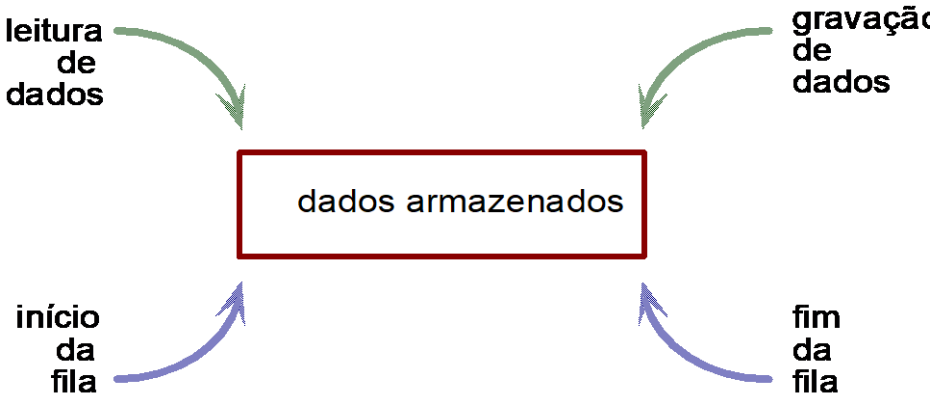


Figura 7.1 Estrutura do tipo fila

Uma Máquina de Post consiste de duas partes:

***Variável X*.**

- T rata-se de uma variável do tipo fila e é utilizada como entrada, saída e memória de trabalho.
- A variável X não possui tamanho nem limite fixos. Seu comprimento é igual ao comprimento da palavra corrente.
- O s símbolos podem pertencer ao alfabeto de entrada ou a { # }, único símbolo auxiliar.
- In icialmente, o valor de X é a palavra de entrada. Caso X não contenha símbolos, a entrada é vazia, representada por ε.

***Programa*.**

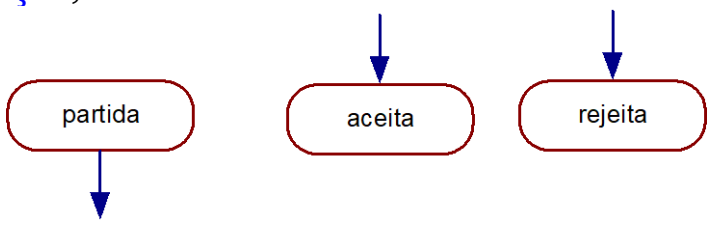
É uma seqüência finita de instruções, representado como um diagrama de fluxos (espécie de fluxograma), no qual cada vértice é uma instrução.
As instruções podem ser de quatro tipos: partida, parada, desvio (leitura com teste) e atribuição.

Definição 7.1

Máquina de Post.

Uma *Máquina de Post* é uma tripla: $M = (\Sigma, D, \#)$ onde:

- Σ *alfabeto de símbolos de entrada*;
- \# *símbolo auxiliar*.
- D *programa* ou *diagrama de fluxos* construído a partir de componentes elementares: partida, parada, desvio e atribuição;



a) Partida. Existe somente uma instrução de início em um programa

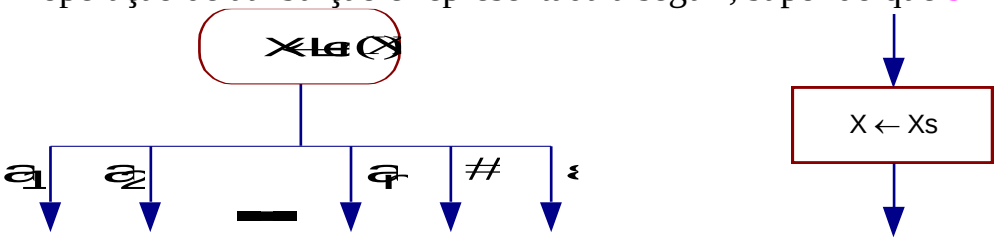
b) Parada. Existem duas alternativas de instruções de parada em um programa, uma de aceitação e outra de rejeição:

c) Desvio (ou leitura com teste). *X□ ler (X)*
denota o comando que lê o símbolo mais à esquerda da palavra armazenada em X, retirando o primeiro símbolo.

- É uma instrução composta de uma leitura do símbolo à esquerda (início da fila), excluindo-o da fila e desviando o fluxo do programa de acordo com o símbolo lido;
- de ve ser prevista a possibilidade de X conter a palavra vazia.
- Po rtanto, é um desvio condicional, e trata-se de uma função total, estando definida para todos os valores do domínio.
- Se o cardinal de Σ é n, então existem n+2 arestas de desvios condicionais, pois se deve incluir as possibilidades # e ε,

d) Atribuição. *X←Xs*

- É uma instrução de concatenação, gravando o símbolo indicado (pertencente a $\sum \cup \{ \# \}$) à direita da palavra armazenada na variável X (fim da fila).
- A operação de atribuição é representada a seguir, supondo que $s \in \sum \cup \{ \# \}$.



Exemplo 7.1
Máquina de Post – Duplo Balanceamento.

Linguagem Duplo_Bal = { $a^n b^n \mid n \geq 0$ }
Máquina de Post: Post_Duplo_Bal = ({ a, b }, D, ?)
onde D é, como ilustrado abaixo, tal que:

ACEITA(Post_Duplo_Bal) = Duplo_Bal
REJEITA(Post_Duplo_Bal) = ∑* - Duplo_Bal
LOOP(Post_Duplo_Bal) = ∅.

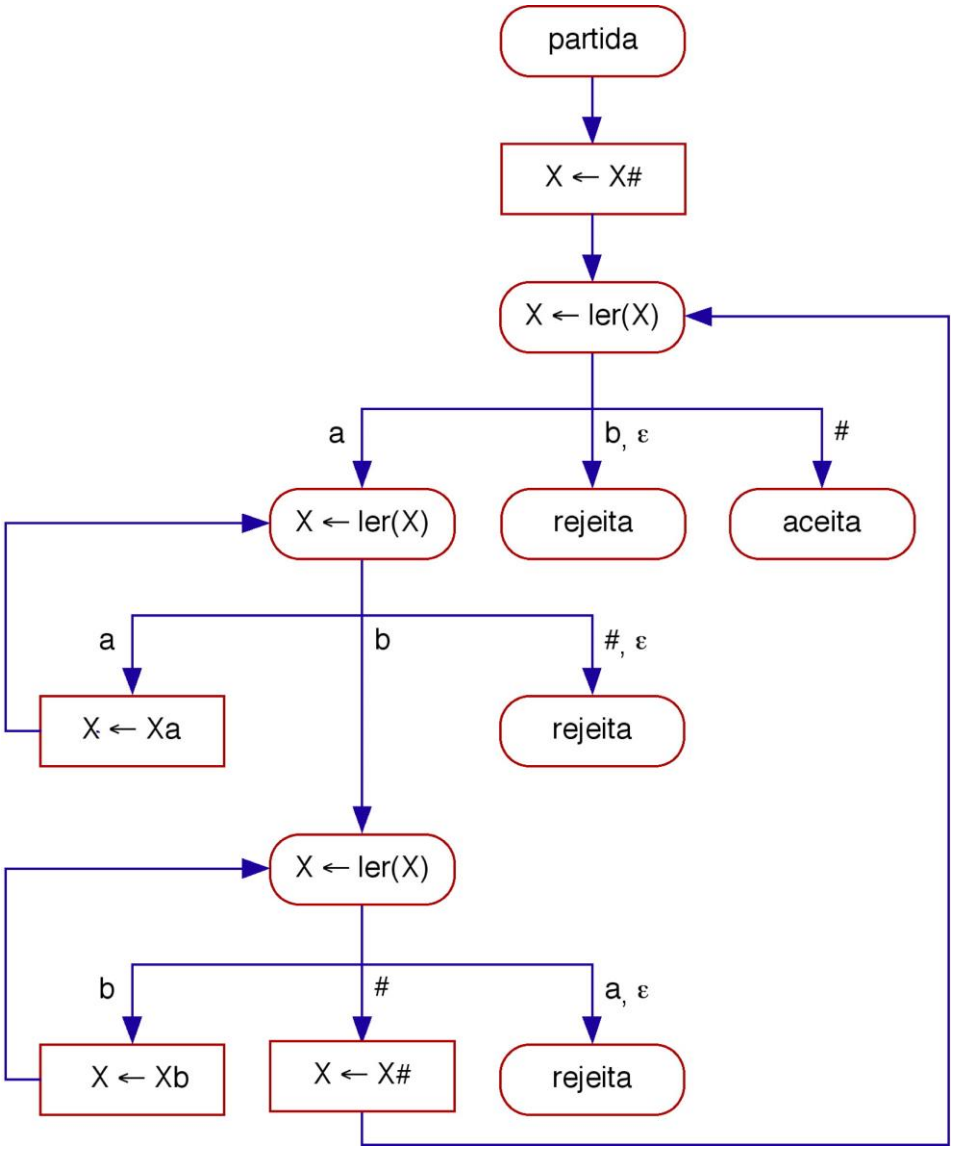


Figura 7.5 Diagrama de Fluxo da Máquina de Post – Duplo-Bal

Exemplo 7.2
Máquina de Post – Duplo-bal- qualquer-ordem

Linguagem **Duplo_Bal_2** = { w | w = $a^n b^n$ ou w = $b^n a^n$, n ≥ 0 }

Máquina de Post: **Post_Duplo_Bal_2** = ({ a, b }, D, ?)

onde D é, como ilustrado abaixo, tal que:

ACEITA(Post_Duplo_Bal_2) = Duplo_Bal_2

REJEITA(Post_Duplo_Ba_2) = ∑* - Duplo_Bal_2

LOOP(Post_Duplo_Bal_2) = ∅.

Figura 7.6 Diagrama de Fluxo da Máquina de Post – $a^n b^n$ ou $b^n a^n$

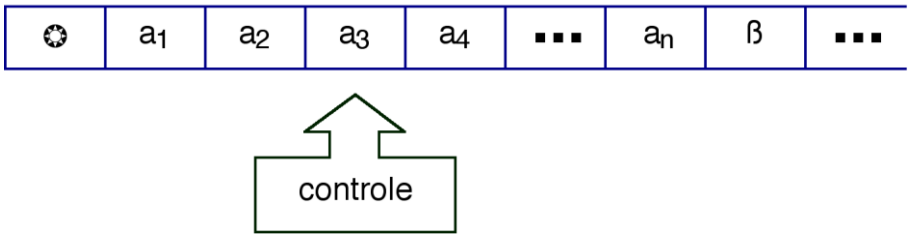
Teorema 7.2
Equivalência entre Máquina de Turing e Máquina de Post.

O formalismo Máquina de Turing pode ser simulado pelo formalismo Máquina de Post e vice-versa.

a) Máquina de Turing ≤ Máquina de Post.

O formalismo Máquina de Turing pode ser simulado pelo formalismo Máquina de Post.

- A estrutura de fita da Máquina de Turing é simulada pela Máquina de Post, usando a variável X, onde a posição corrente da cabeça corresponde à primeira posição em X.



X = a3 a4 ... an # a1 a2

Figura 7.7 Fita e posição da cabeça da fita

- o movimento para a esquerda da cabeça da fita, exige a execução de tantas instruções desvio (leitura com teste) e atribuições quantos forem os símbolos da palavra corrente;
- o movimento para a direita da cabeça da fita, consiste na execução da instrução desvio, onde ocorre a leitura de um símbolo da variável X (no início da fila) e a correspondente **atribuição** (no fim da fila);

a simulação dos estados é como segue:

- estado inicial é simulado pela instrução partida, seguida pela execução de uma instrução desvio;
- os estados finais são simulados pela instrução aceita;
- para cada um dos demais estados, corresponde a uma instrução desvio (teste);
- as condições de rejeição (função programa indefinida ou movimento inválido) são simuladas pela instrução rejeita;

b) Máquina de Post ≤ Máquina de Turing

O formalismo Máquina de Post pode ser simulado pelo formalismo Máquina de Turing.

- A variável X da Máquina de Post é simulada na Máquina de Turing, usando a estrutura de fita, onde a primeira posição da fila corresponde à posição corrente da cabeça da fita.

$X = a_1\ a_2\ a_3 \ldots a_m\ \#\ a_{m+1} \ldots a_n$

Figura 7.8 Simulação da fila pela fita

- **o** desvio realiza a leitura com remoção do símbolo mais à esquerda da variável X, o que pode ser simulado pela substituição do conteúdo da correspondente célula de fita pelo símbolo branco, resultando em símbolos brancos entre o marcador de inicio de fita e primeiro símbolo não-branco (no caso de inconveniência desses brancos, pode-se realizar um deslocamento do conteúdo para a esquerda);
- a **atribuição** de um símbolo s (à direita do conteúdo da variável X), pode ser simulada movendo a cabeça para o fim da fita, gravando o símbolo s e retornar para a posição correspondente ao primeiro símbolo da fila;
- a instrução partida pode ser simulada usando o estado inicial;
- uma instrução aceita pode ser simulada usando um estado final;
- uma instrução rejeita pode ser simulada usando uma condição excepcional de parada (como um movimento inválido).

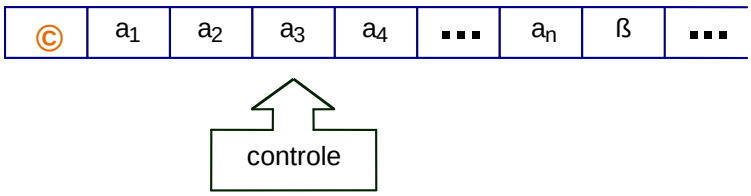
a) PROVA: Turing ≤ Post

- Seja a Máquina de Turing $M = (\Sigma, Q, \Pi, q_0, F, V, ß, ©)$.
- A simulação por uma Máquina de Post $M' = (\Sigma \cup V, D, \#)$

pode ser definida como segue:

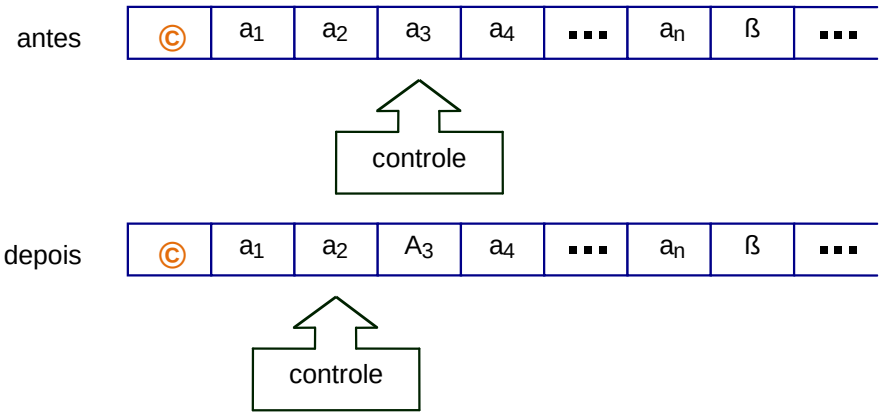
a) *Fita.*

A fita é simulada pela variável X, e a posição corrente da cabeça da fita é representada pela primeira posição da fila (ou variável); o símbolo especial # é introduzido para indicar na variável X o que estava à esquerda da cabeça da fita, a partir do início da fita;



$X = a_3\ a_4\ \ldots\ a_n\ \#\ a_1\ a_2$

b) *Movimento para a Esquerda.*



Se o conteúdo da variável X antes do movimento é o seguinte:

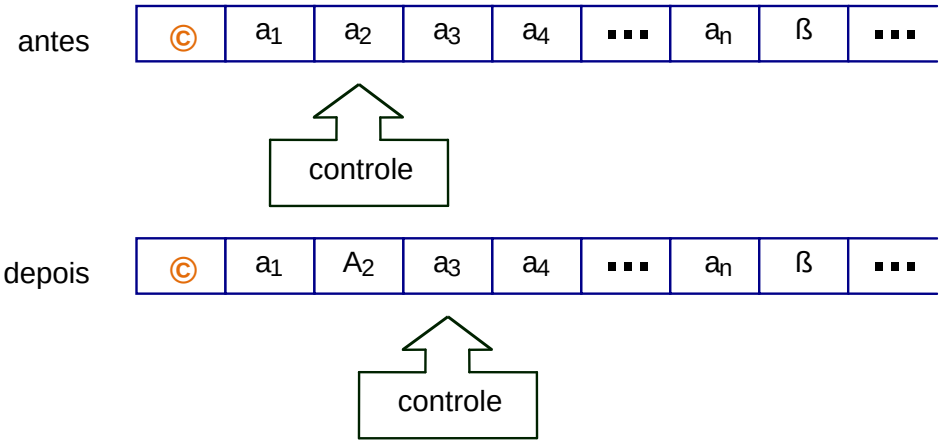
$X = a_3\ a_4\ \ldots\ a_n\ \#\ a_1\ a_2$

Simular o movimento para a esquerda da cabeça e a substituição do símbolo a_3 por A_3, é necessário alterar o conteúdo de X como segue:

$X = a_2\ A_3\ a_4\ \ldots\ a_n\ \#\ a_1$

Para tal são necessários tantos testes (desvio) e **atribuições** quantos forem os símbolos da palavra corrente.

c) Movimento para a Direita.



Conteúdo da variável X antes do movimento é

$$X = a_2\ a_3\ a_4\ \ldots\ a_n\ \#\ a_1$$

Para simular o movimento para a direita da cabeça é necessário alterar o conteúdo de X como segue, o que é trivial:

$$X = a_3\ a_4\ \ldots\ a_n\ \#\ a_1\ A_2$$

d) Estados.

A simulação dos estados é como segue:

- *Estado Inicial*. Simulado pela instrução partida;
- *Estados Finais*. Simulados pela instrução aceita;
- *Demais Estados*. Cada estado corresponde a uma instrução desvio (leitura com teste);

e) Condições de Rejeição.

As condições de rejeição da Máquina de Turing (função programa indefinida ou movimento inválido) são simuladas em Post por rejeita.

Esta Máquina de Post, construída conforme esta definição simulará a Maquina de Turing dada!

b) PROVA: *Máquina de Post* ≤ *Máquina de Turing*

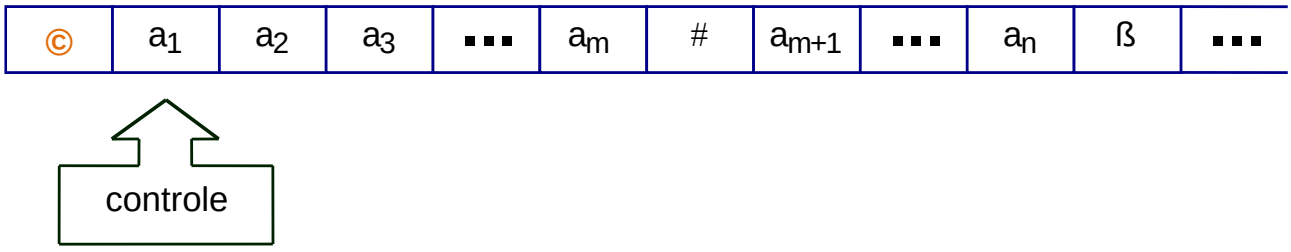
- Suponha uma Máquina de Post M = (∑, D, #).
- A simulação de M por uma Máquina de Turing

  M' = (∑, Q, Π, q_0, F, { # }, ß, ©) pode ser definida por:

a) *Variável X*.

A variável X é simulada pela fita, e a posição mais à esquerda da fila é representada pela posição da cabeça da fita.

Para **X** = a_1 a_2 a_3 ... a_m #a_{m+1} ... a_n



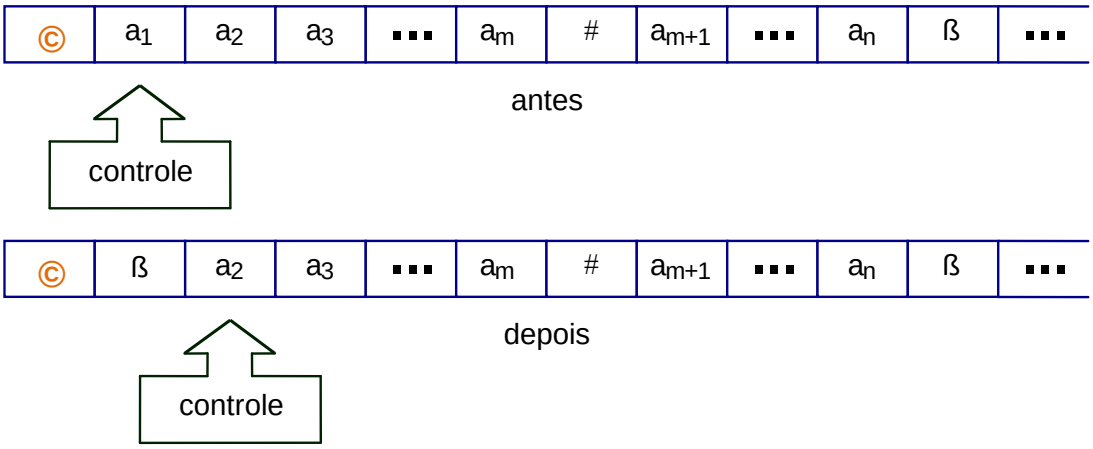
b) Desvio. *X← ler(X)*.

Se o conteúdo da variável X é: **X = a1 a2 a3 ... am #am+1 ... an**

A leitura e remoção do símbolo mais à esquerda resulta em:

X = a2 a3 ... am #am+1 ... an

Isso pode ser simulado pela alteração da fita e cabeça da fita



c) Atribuição. *X←Xs*.

A concatenação de um símbolo **s** deve sempre ser à direita do conteúdo da variável X (ou seja, no fim da fila).

Para o conteúdo de X

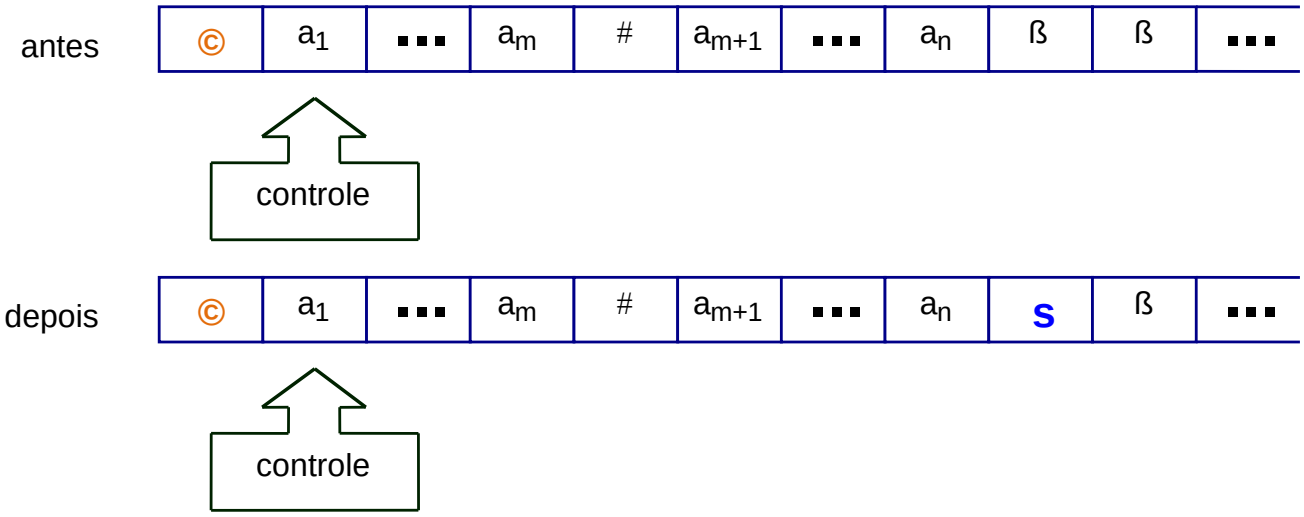
$$X = a_1 \ldots a_m \#a_{m+1} \ldots a_n$$

resulta em

$$X = a_1 \ldots a_m \#a_{m+1} \ldots a_n \mathbf{s}$$

o que pode ser simulado pela Máquina de Turing:

> Move-se a cabeça para o fim da fita, grava-se o símbolo **s** e retorna-se para a posição correspondente ao primeiro símbolo da fila.



d) *Partida*.

A instrução partida pode ser simulada em uma Máquina de Turing usando o estado inicial

e) *Aceita*.

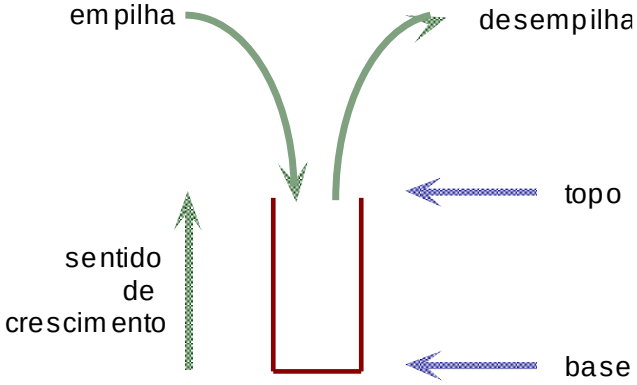
Uma instrução aceita pode ser simulada em uma Máquina de Turing usando um estado final

f) *Rejeita*.

Uma instrução rejeita pode ser simulada em uma Máquina de Turing usando uma condição excepcional de parada (como um movimento inválido).

7.2 Máquina com Pilhas

- A principal característica é o uso da estrutura do tipo pilha como memória de trabalho.
- São necessárias **duas** pilhas para que a Máquina seja Universal.



Definição 7.3

Máquina com Pilhas.

Uma *Máquina com Pilhas* é uma dupla:

M = (∑, D) onde:

- ∑ *alfabeto de símbolos de entrada*;
- D *programa* ou *diagrama de fluxos* construído a partir de componentes elementares denominados partida, parada, desvio, empilha e desempilha.

Uma *Máquina com Pilhas* consiste de três partes:

a) *Variável X*. É usada somente para entrada, não possui tamanho nem limite fixos; seu comprimento é igual ao comprimento da palavra corrente armazenada. Caso X não contenha símbolos, a entrada é vazia, representada por ε.

b) *Variáveis* Y_i. São do tipo pilha e utilizadas como memória de trabalho; não possuem tamanho nem limite fixos; Inicialmente, o valor de cada pilha Y_i é a palavra vazia, e seu comprimento é igual ao comprimento da palavra corrente armazenada.

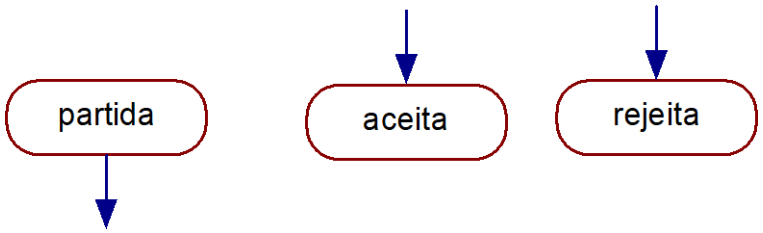
c) Programa. É uma seqüência finita de *instruções*, representado como um diagrama de fluxos onde cada vértice é uma instrução.

- A s instruções podem ser de cinco tipos: partida, parada, desvio, desempilha e empilha.

As componentes elementares de um diagrama de fluxos são:

a) *Partida*. Existe somente uma instrução de início (partida) em um programa.

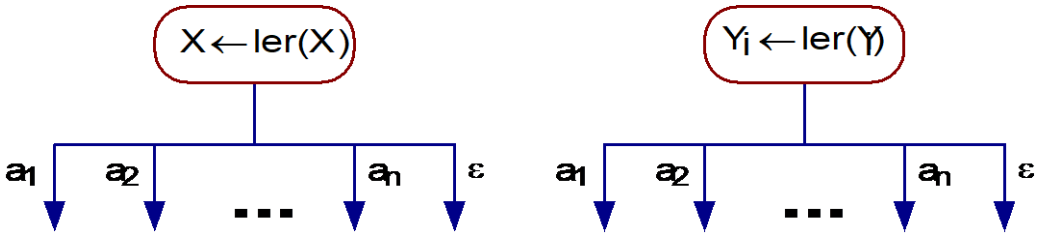
b) *Parada*. Existem duas alternativas de instruções de parada em um programa: uma de aceitação (aceita), e outra de rejeição (rejeita).



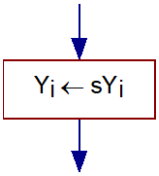
c) *Desvio* (ou *Teste*) e *Desempilha*. Determinam o fluxo do programa de acordo com o símbolo mais à esquerda da palavra armazenada na variável X (desvio) ou no topo da pilha Y_i (desempilha).

São desvios condicionais. Se o cardinal de $\sum$ é n, então existem n+1 arestas de desvios condicionais, pois se deve incluir a possibilidade da palavra vazia $\square$.

$X \leftarrow ler(X)$ denota uma *leitura destrutiva*, que lê o símbolo mais à esquerda de X ou do topo de Y_i, retirando o símbolo lido.



d) *Empilha*. Empilha um símbolo $s \in \sum$ no topo da pilha indicada, ou seja, concatena o símbolo na extremidade da palavra armazenada na variável Y_i.


$Y_i \leftarrow sY_i$

- em um diagrama de fluxos, existe somente uma instrução de partida, mas podem existir diversas (zero ou mais) instruções de parada, tanto de aceitação como de rejeição;
- uma palavra de entrada é aceita ou rejeitada, se a computação, iniciada com a variável X contendo a entrada, atingir uma instrução aceita ou rejeita, respectivamente;
- em um desvio (respectivamente, desempilha), se X (respectivamente, Y_i) contém a palavra vazia ε, então segue o fluxo correspondente; caso contrário, lê o símbolo mais à esquerda de X (respectivamente, no topo de Y_i) e o remove após a decisão de qual aresta do fluxo indica a próxima instrução.

As seguintes conclusões podem ser estabelecidas sobre o número de pilhas e o poder computacional das máquinas com pilhas:

a) ***Máquina Finita***. Uma Máquina Finita, que corresponde a uma *Máquina Sem Pilhas*, possui um poder computacional relativamente restrito, pois não tem memória auxiliar para armazenar informações de trabalho. Por exemplo, não existe Máquina Finita capaz de reconhecer um duplo balanceamento como em $\{ a^n b^n \mid n \geq 0 \}$.

b) ***Máquina com Uma Pilha***. A Classe das Máquinas com Uma Pilha, embora mais poderosa que a Classe das Máquinas Finitas, ainda possui uma capacidade computacional restrita. Por exemplo, não existe Máquina com Uma Pilha capaz de reconhecer um triplo balanceamento como em $\{ a^n b^n c^n \mid n \geq 0 \}$.

c) ***Máquina com Duas Pilhas***. Conforme será verificado adiante, a Classe das Máquinas com Duas Pilhas possui o mesmo poder computacional que a Classe das Máquinas de Turing ou de Post;

d) ***Máquina com Mais de Duas Pilhas***. A Classe das Máquinas com Mais de Duas Pilhas possui o mesmo poder computacional que a Classe das Máquinas com Duas Pilhas. Ou seja, para qualquer máquina com mais de duas pilhas, é possível construir uma equivalente com, no máximo, duas pilhas que realiza o mesmo processamento.

Exemplo 7.3
Máquina com Pilhas – Duplo Balanceamento.

Considere a seguinte linguagem

$$Duplo\_Bal = \{ a^n b^n \mid n \geq 0 \}$$

A Máquina com Pilhas:

$$Pilhas\_Duplo\_Bal = (\{ a, b \}, D)$$

onde D é tal que:

ACEITA(Pilhas_Duplo_Bal) = Duplo_Bal
REJEITA(Pilhas_Duplo_Bal) = ∑* - Duplo_Bal
LOOP(Pilhas_Duplo_Bal) = ∅

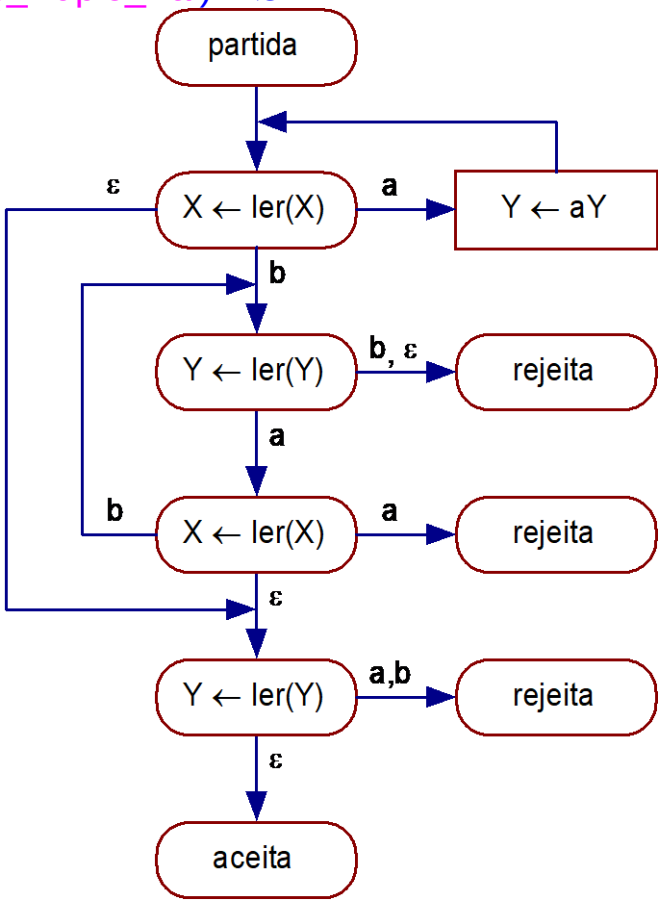


Figura 7.14
Diagrama de Fluxo da Máquina com UMA Pilha – Duplo- Bal

Descrição do Programa:

O algoritmo lê o prefixo de símbolos a e empilha na única pilha utilizada Y. Após, para cada símbolo b em X deve existir um correspondente a em Y.

Exemplo 7.4
Máquina com Duas Pilhas – Triplo Balanceamento.

Considere a seguinte linguagem

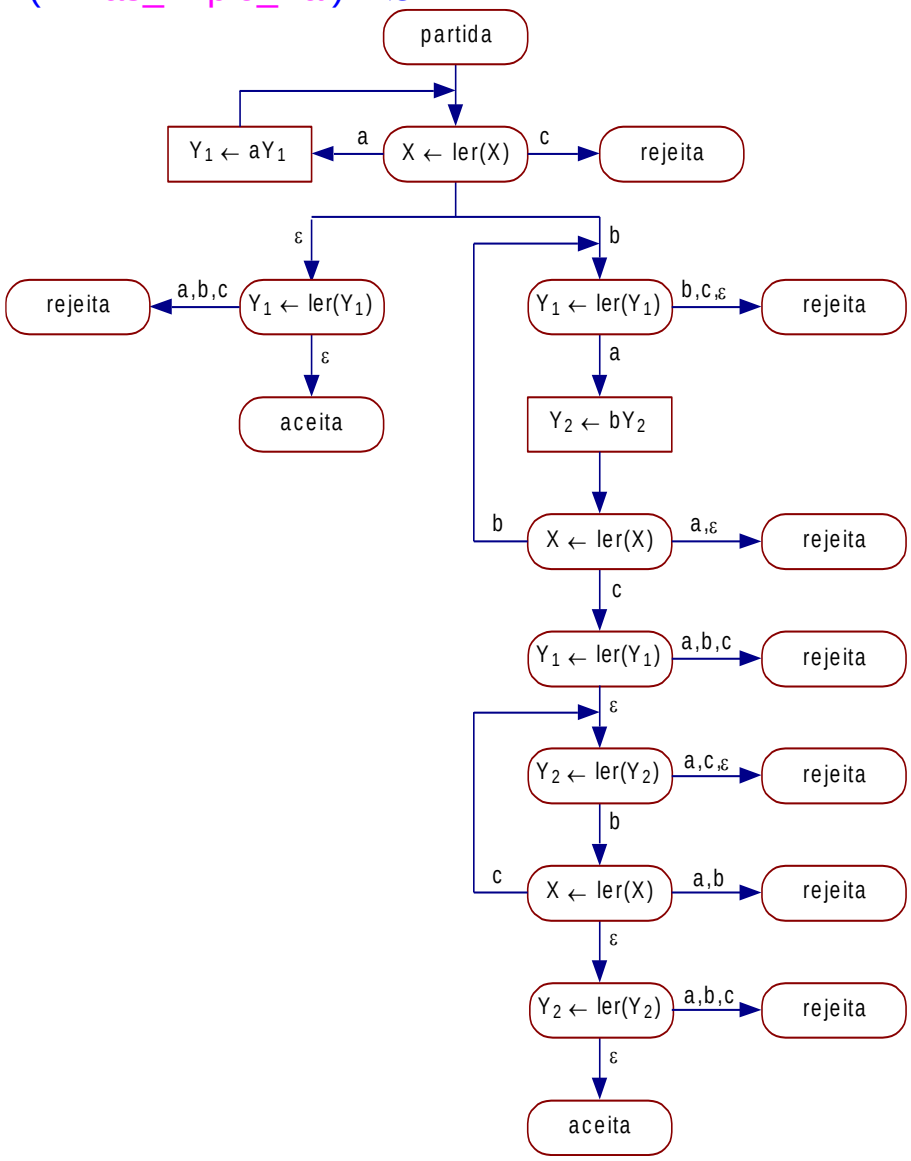
$$Triplo\_Bal = \{ a^n b^n c^n \mid n \geq 0 \}$$

Máquina com Pilhas:

$$Pilhas\_Triplo\_Bal = (\{ a, b, c \}, D)$$

onde D é tal que:

$$ACEITA(Pilhas\_Triplo\_Bal) = Triplo\_Bal$$
$$REJEITA(Pilhas\_Triplo\_Bal) = \Sigma^* - Triplo\_Bal$$
$$LOOP(Pilhas\_Triplo\_Bal) = \varnothing$$



Exemplo 7.5
Máquina Finita – Prefixo.

Considere a seguinte linguagem:

Prefixo_aaa={w | w∈{a,b}* e w possui a subpalavra aaa como prefixo}

Máquina:

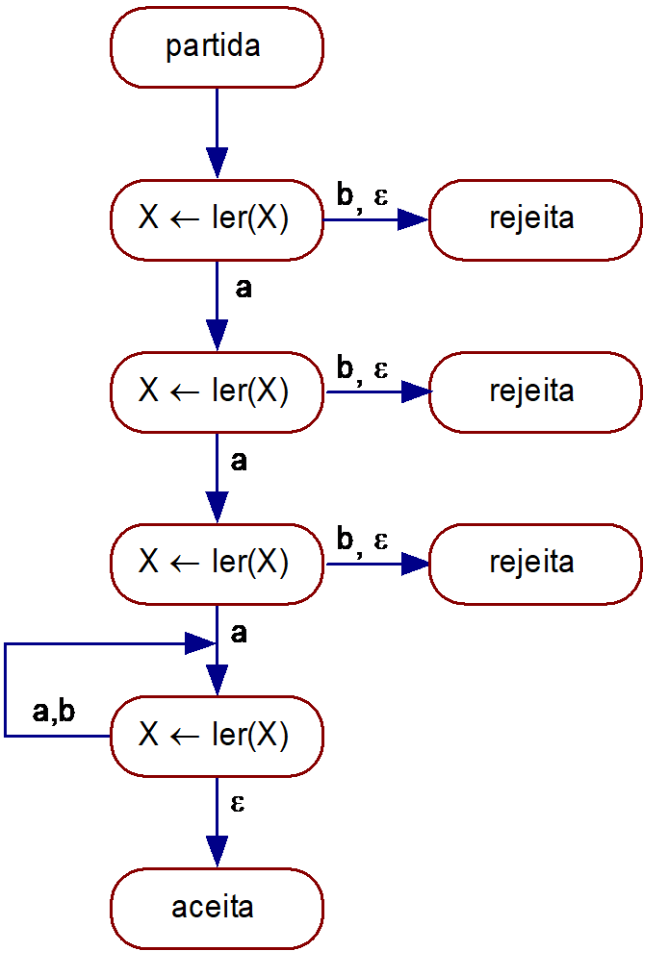
finita_Prefixo_aaa = ({ a, b }, D)

onde D é tal que:

ACEITA(finita_Prefixo_aaa) = Prefixo_aaa

REJEITA(finita_Prefixo_aaa) = ∑* - Prefixo_aaa

LOOP(finita_Prefixo_aaa) = ∅



7.3 Autômato com Pilhas

- *Autômato com Pilhas* ou *Autômato com Duas Pilhas* possui a memória de entrada separada das memórias de trabalho e de saída.
- Essas estruturas auxiliares são do tipo pilha, e cada máquina possui duas pilhas.

Um Autômato com Pilhas é composto por:

a) *Fita*. Dispositivo de entrada que contém a informação a ser processada;

b) *Duas Pilhas*. Memórias auxiliares que podem ser usadas livremente para leitura e gravação;

➢ Uma *pilha* é dividida em células, armazenando, cada uma, um símbolo do alfabeto auxiliar (pode ser igual ao alfabeto de entrada). Em uma estrutura do tipo pilha, a leitura ou gravação é sempre na mesma extremidade (topo). Não possui tamanho fixo e nem máximo, sendo seu tamanho corrente igual ao tamanho da palavra armazenada. Seu valor inicial é vazio.

c) *Unidade de Controle*. Reflete o estado corrente da máquina. Possui uma cabeça de fita e uma cabeça para cada pilha;

➢ A *unidade de controle* possui um número finito e predefinido de estados. Possui uma cabeça de fita e uma cabeça para cada pilha:

⇒ *Cabeça da Fita*. Unidade de leitura que acessa uma célula da fita de cada vez e movimenta-se exclusivamente para a direita.

⇒ *Cabeça da Pilha*. Unidade de leitura e gravação para cada pilha que move para cima ao gravar e para baixo ao ler um símbolo. Acessa um símbolo de cada vez, estando sempre posicionada no topo. A leitura exclui o símbolo lido.

d) *Programa ou Função de Transição*. Comanda a leitura da fita, a leitura e gravação das pilhas e define o estado da máquina.

➢ *programa* é uma função parcial que, dependendo do estado corrente, símbolo lido da fita e do símbolo lido de cada pilha, determina o novo estado e o símbolo a ser gravado em cada pilha.

Definição 7.4

Autômato com Duas Pilhas.

Um *Autômato com Duas Pilhas* ou simplesmente *Autômato com Pilhas* M é uma 6-upla:

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

onde:

- Σ *alfabeto de símbolos de entrada*;
- Q *conjunto de estados* possíveis do autômato, o qual é finito;
- Π *função programa* ou de *função de transição*:

$$\Pi: Q \times (\Sigma \cup \{\varepsilon, ?\}) \times (V \cup \{\varepsilon, ?\}) \times (V \cup \{\varepsilon, ?\}) \rightarrow Q \times (V \cup \{\varepsilon\}) \times (V \cup \{\varepsilon\})$$

  a qual é uma função parcial;
- q_0 *estado inicial* do autômato, tal que q_0 é elemento de Q;
- F *conjunto de estados finais,* tal que F está contido em Q;
- V *alfabeto auxiliar*.

As seguintes características da função programa devem ser consideradas:

- *a função pode não ser total, ou seja, pode ser indefinida para alguns argumentos do conjunto de partida; a omissão do parâmetro de leitura, representada por "?", indica o teste da correspondente pilha vazia ou de toda a palavra de entrada lida;*
- *o símbolo ε na leitura da fita ou de alguma pilha indica que o autômato não lê nem move a cabeça. Note-se que, pelo menos uma leitura deve ser realizada ou sobre a fita ou sobre alguma pilha;*
- *o símbolo ε na gravação indica que nenhuma gravação é realizada na pilha (e não move a cabeça).*

Função programa *considera*:

- estado corrente;
- símbolo lido da fita (pode ser omitido) ou teste se toda a palavra de entrada foi lida;
- símbolo lido de cada pilha (pode ser

determina

- novo estado;
- símbolo gravado em cada pilha (pode ser omitido).

omitido) ou teste de pilha vazia;

Π(p, ?, a, ε) = { (q, ε, b) } indica que se:

- no estado p;
- a entrada foi completamente lida (na fita);
- o topo da pilha 1 contém o símbolo a;
- não lê da pilha 2;
- assume o estado q;
- não grava na pilha 1;
- grava o símbolo b no topo da pilha 2.

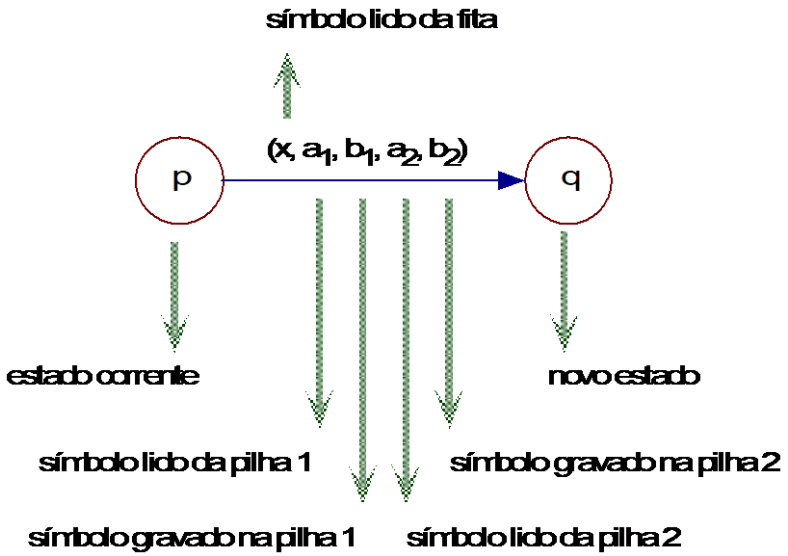


Figura 7.15 Representação da função programa como um grafo

- *processamento de um Autômato com Pilhas*, para uma palavra de entrada w, consiste na sucessiva aplicação da função programa para cada símbolo de w (da esquerda para a direita) até ocorrer uma condição de parada.
- Um exemplo simples de *ciclo infinito* é um programa que empilha e desempilha um mesmo símbolo indefinidamente, sem ler da fita.
- As *condições de parada* são as seguintes:
  - *Estado Final*. O autômato assume um estado final: o autômato pára, e a palavra de entrada é aceita;
  - *Função Indefinida*. A função programa é indefinida para o argumento: o autômato pára, e a palavra de entrada é rejeitada.

Exemplo 3.25
Autômato com Pilhas – Duplo-Bal

Considere a linguagem

$Duplo\_Bal = \{ a^n b^n \mid n \geq 0 \}$

O Autômato com Pilhas

$A2P\_Duplo\_Bal = (\{ a, b \}, \{ q_0, q_1, q_f \}, \Pi, q_0, \{ q_f \}, \{ B \})$

onde Π é tal que

$\Pi(q_0, a, \varepsilon, \varepsilon) = (q_0, B, \varepsilon)$
$\Pi(q_0, b, B, \varepsilon) = (q^1, \varepsilon, \varepsilon)$
$\Pi(q_0, ?, ?, \varepsilon) = (q_f, \varepsilon, \varepsilon)$
$\Pi(q_1, b, B, \varepsilon) = (q_1, \varepsilon, \varepsilon)$
$\Pi(q_1, ?, ?, \varepsilon) = (q_f, \varepsilon, \varepsilon)$

$ACEITA(A2P\_Duplo\_Bal) = Duplo\_Bal$

Descrição do Programa:

- a utômato pode ser representado por um grafo.
- N ão foi utilizada a segunda pilha em nenhum momento.
- N o estado q_0, para cada símbolo a lido da fita, é armazenado um símbolo B na pilha 1.
- N o estado q_1, é realizado um batimento, verificando se, para cada símbolo b da fita, existe um correspondente B na pilha 1.
- O algoritmo somente aceita se, ao terminar de ler toda a palavra de entrada, a pilha 1 estiver vazia.

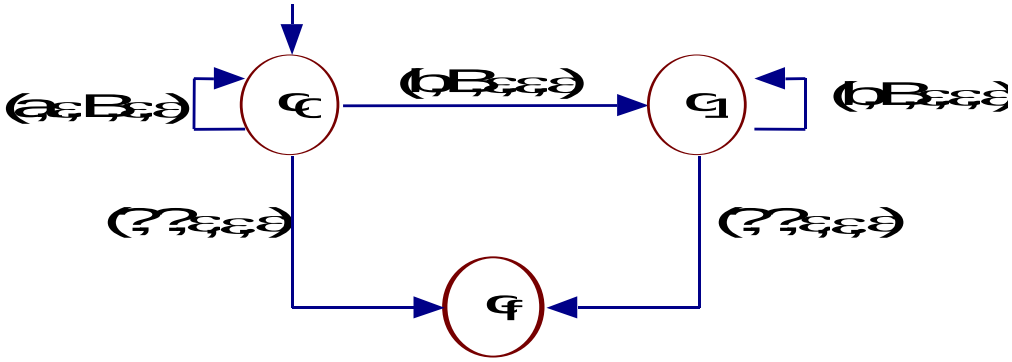


Figura 7.16 Grafo do Autômato com Pilhas - Duplo Balanceamento

Exemplo 7.7
Autômato com pilhas – triplo balanceamento.

Considere a linguagem:

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

O Autômato com Pilhas

$$A2P\_Triplo\_Bal = (\{ a, b, c \}, \{ q_0, q_1, q_2, q_f \}, \Pi, q_0, \{ q_f \}, \{ B, C \})$$

é tal que $ACEITA(A2P\_Triplo\_Bal) = Triplo\_Bal$ (e sempre pára).

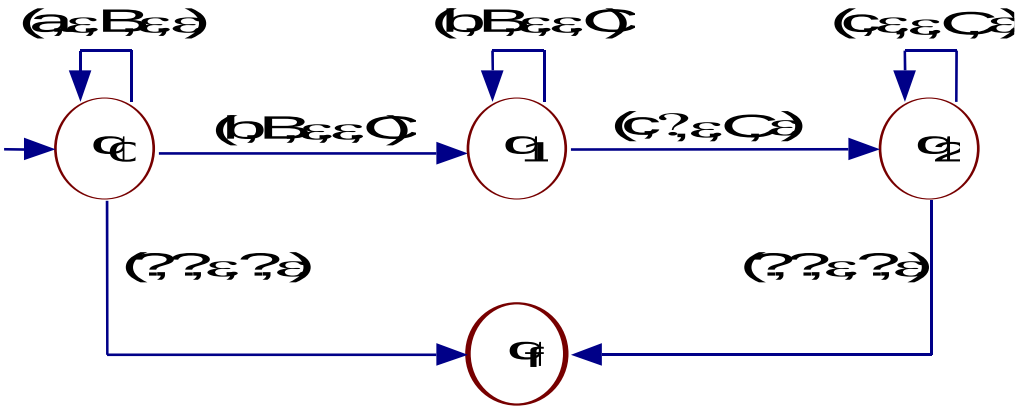


Figura 7.17 Grafo do Autômato Finto com Pilhas – Triplo-Bal

Descrição do Programa:

- No estado q_0, para cada símbolo a lido da fita, é armazenado um símbolo B na pilha 1.
- No estado q_1, é realizado um batimento, verificando se, para cada símbolo b da fita, existe um correspondente B na pilha 1, bem como é armazenado na pilha 2 um símbolo C.
- Por fim, no estado q_2, é realizado um batimento, verificando se, para cada símbolo c da fita, existe um correspondente C na pilha 2.

- O algoritmo somente aceita se, ao terminar de ler toda a palavra de entrada, as pilhas estiverem vazias.

Teorema 7.5
Equivalência entre Máquina de Turing e Autômato com Duas Pilhas

O formalismo Máquina de Turing pode ser simulado pelo formalismo Autômato com Duas Pilhas e vice-versa.

a) *Máquina de Turing ≤ Autômato com Duas Pilhas*.

- A estrutura de fita da Máquina de Turing é simulada usando as duas pilhas como segue: a pilha 1 simula o conteúdo da fita à esquerda da cabeça da fita, e a pilha 2, o conteúdo à direita;

b) *Autômato com Duas Pilhas ≤ Máquina de Turing* .

- A fita e as duas pilhas do Autômato com Duas Pilhas são simulada usando a fita da Máquina de Turing, como segue:
  - a palavra de entrada corresponde às primeiras posições da fita da Máquina de Turing;
  - a pilha 1 corresponde as células ímpares da fita, após a palavra de entrada;
  - a nalogamente, a pilha 2 corresponde às células pares da fita, após a palavra de entrada.

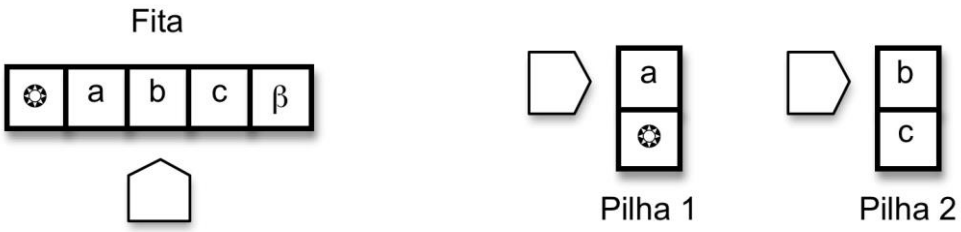


Figura 7.18 Simulação da fita por duas pilhas

7.4 Máquinas Não-Determinísticas

- ***não-determinismo*** *é uma importante generalização dos modelos de máquinas.*
- A mesma idéia é válida para a variável X da Máquina de Post ou para as pilhas do Autômato com Pilhas.

Genericamente, *não-determinismo* é interpretado como:

- a máquina, ao processar uma entrada, tem como resultado um conjunto de novos estados.
- el a assume um conjunto de estados alternativos, como se houvesse uma multiplicação da unidade de controle, uma para cada alternativa, processando independentemente, sem compartilhar recursos com as demais.
- pr ocessamento de um caminho não influi no estado geral, nem no símbolo lido dos demais caminhos alternativos.

Para uma máquina M Não-determinística, uma palavra w pertence a:

- ACEITA(M) se existe pelo menos um caminho alternativo que aceita a palavra.
- REJEITA(M) se todas as alternativas rejeitam a entrada.
- LOOP(M) se nenhum caminho aceita a palavra e pelo menos um fica em *loop*.

Teorema 7.6
Equivalência entre Máquina de Turing e Máquinas Não-Determinísticas

O formalismo Máquina de Turing pode ser simulado pelos seguintes formalismos e vice-versa.

a) Máquina de Turing Não-Determinística;
b) Máquina de Post Não-Determinística;
c) Máquina com Pilhas Não-Determinística.

d) Autômato com Pilhas Não-Determinístico.

Exemplo 7.8
Autômato com Pilhas Não-Determ Palavra & Reversa

Considere a seguinte linguagem:

Palavra_Reversa = { ww^r | w pertence a { a, b }* }

A linguagem Palavra_Reversa contém todas as palavras sobre o alfabeto { a, b } tais que a primeira metade é igual à segunda metade, mas invertida.

Exemplos: □, **abbbba, abaabaabaaba, bbbbbbbbaabbbbbbbb**

Autômato com Pilhas APN_Palavra_Reversa =
ACEITA(APN_Palavra_Reversa)= Palavra_Reversa

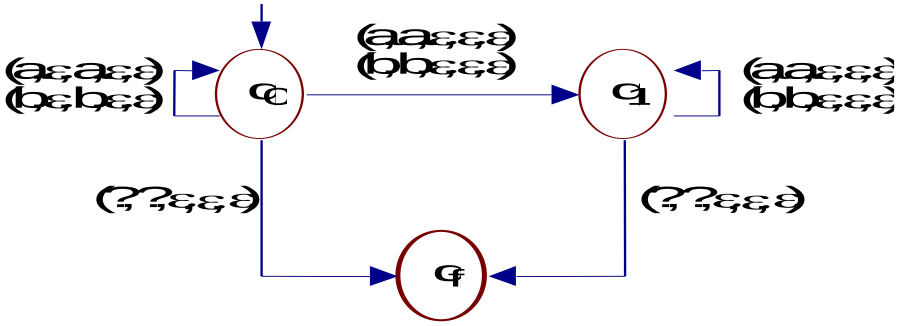
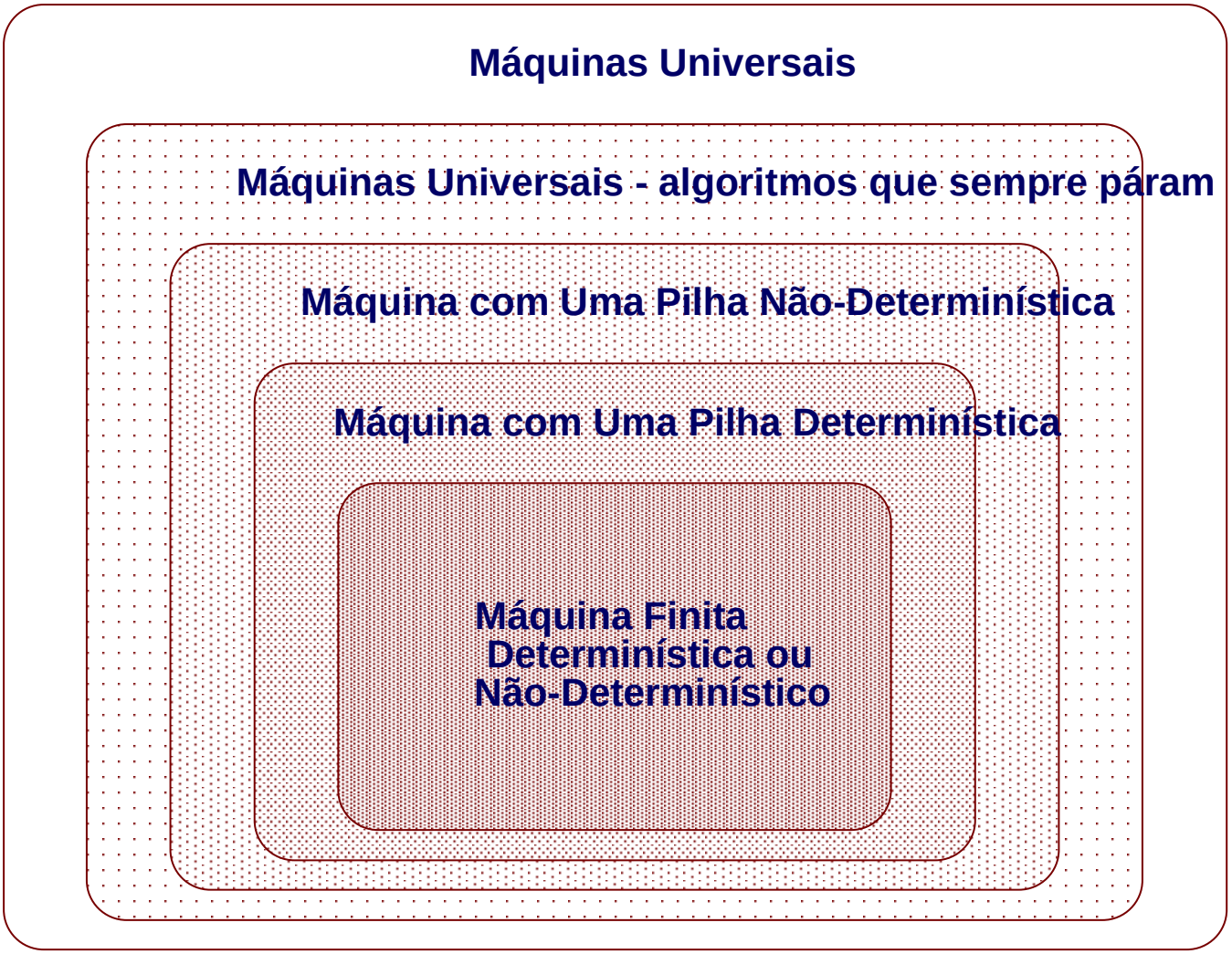


Figura 7.19
Grafo do Autômato com Pilhas Não-Determ – palavra e sua reversa

Descrição do Programa:

- É não-determinístico devido às duas alternativas de movimentos a partir de q_0, para os mesmo símbolos lidos da fita de entrada (ciclo em q_0 e desvio para q_1).
- Adicionalmente, o alfabeto auxiliar é igual ao de entrada. Em q_0, é empilhado (pilha 1) o reverso do prefixo.
- A cada símbolo empilhado, se existe o correspondente símbolo no topo da pilha 1, então:
  - o autômato permanece no estado q_0 e continua empilhando o reverso da entrada (pois não existe controle se já identificou toda a primeira metade);
  - ocorre um movimento não-determinista para q_1 (e, portanto, o autômato inicia uma alternativa) o qual verifica se o sufixo da palavra é igual ao conteúdo da pilha 1 até então empilhado.

7.5 Hierarquia de Classes de Máquinas



Hierarquia de Classes de Linguagens:

a) *Linguagens Regulares*. Correspondem à Classe das Máquinas sem Pilha ou Finitas.

- São exemplos de linguagens dessa classe: alguns editores de texto, protocolos de comunicação.
- São linguagens muito simples, como as quais, por exemplo, não é possível fazer qualquer tipo de balanceamento de tamanho não-predefinido.
- A principal característica dessa classe é que o tempo de reconhecimento de uma palavra é diretamente proporcional ao comprimento da entrada.
- qualquer algoritmo especificado usando esse formalismo é igualmente eficiente, em termos de tempo de processamento;

b) *Linguagens Livres do Contexto Determinísticas*. Correspondem à Classe dos Máquinas ou Automatos Determinísticos com Uma Pilha.

- São linguagens mais complexas que as Regulares, mas ainda muito simples, com as quais, por exemplo, não é possível reconhecer a linguagem

  $Palavra\_Reversa = \{ ww^r \mid w \text{ pertence a } \{ a, b \}^* \}$

- tempo de reconhecimento de uma entrada é diretamente proporcional ao dobro do tamanho da entrada;

c) *Linguagens Livres do Contexto*. Correspondem à Classe das Máquinas ou Autômatos Não-Determinísticos com Uma Pilha:

- Constituem uma classe de fundamental importância, pois incluem linguagens de programação como Algol e Pascal.
- Algumas linguagens muito simples não pertencem a essa classe de linguagens como:

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

  $Palavra\_Palavra = \{ ww \mid w \text{ é palavra sobre os símbolos } a \text{ e } b \}$

- Os melhores algoritmos de reconhecimento conhecidos possuem tempo de processamento proporcional ao tamanho da entrada elevado ao cubo;

d) *Linguagens Recursivas*.

- Correspondem à classe de todas as linguagens que podem ser reconhecidas mecanicamente e para as quais existe um algoritmo de reconhecimento que sempre pára para qualquer entrada.
- Inclui a grande maioria das linguagens aplicadas. Os reconhecedores de linguagens recursivas podem ser muito ineficientes, tanto em termos de tempo de processamento como de recursos de memória

e) *Linguagens Enumeráveis Recursivamente*. Correspondem à Classe das Máquinas Universais.

- Correspondem à classe de todas as linguagens que podem ser reconhecidas mecanicamente.
- Os complementos destas linguagens nem sempre podem ser reconhecidas mecanicamente..

Figura 7.21 Hierarquia de Classes de Linguagens

7.6 Conclusões

Neste capítulo foram estudados os formalismos:

- Máquina de Post (baseado em uma variável do tipo fila);
- Máquinas com Pilhas, constituídas por uma família de máquinas com capacidades ou poder computacional distintos. Tem-se a Máquina Finita (sem pilha alguma de memória auxiliar), Máquina com uma ou com duas (ou mais pilhas). Duas pilhas são suficientes para atingir o poder computacional equivalente da Máquina de Turing;
- Autômatos com Pilhas, análogos as Máquinas com Pilhas, diferenciando-se na forma de expressar a função programa, substituindo diagramas de fluxos por grafos diretos, representando estados e transições;
- Máquinas Não-Determinísticas cujos efeitos não aumentam o poder computacional de suas correspondentes versões determinísticas, excetuando-se as Máquinas ou Autômatos com uma pilha.

No próximo capítulo serão vistos outros formalismos equivalentes, mas do tipo denotacional, baseados no conceito de recursão.

7.7 Exercícios

Exercício 7.1

Faça um quadro comparativo entre os modelos Máquina de Turing, Máquina de Post e Autômato com Duas Pilhas, destacando suas características e seus principais aspectos funcionais. Em particular, compare as estruturas de dados de cada modelo.

Exercício 7.2

Desenvolva Máquinas de Post, determinísticas ou não, 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 $\{ a, b \}^*$ e w^r denota a palavra reversa $\}$

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 $j = k \}$

Exercício 7.3

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. Construa uma Máquina de Post sobre $\Sigma = \{ v, f, e, ou \}$ tal que:

ACEITA(M) = { EB | EB é v }

REJEITA(M) = { EB | EB é f }

LOOP(M) = { w | w não é EB }

Exercício 7.4

Dê a Máquina de Post MP, sobre o alfabeto { a, b }, tal que:

ACEITA(MP) = { w | w tem o mesmo número de símbolos a e b }

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

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

Por exemplo:

ab ∈ ACEITA(MP)

aba ∈ REJEITA(MP)

aaba ∈ LOOP(MP).

Exercício 7.5

Elabore uma Máquina de Post P sobre o alfabeto { a, b } tal que:

ACEITA(P) = { $a^m b^n$ | $m > n \geq 0$ }

REJEITA(P)= { a, b }* - ACEITA(P)

Exercício 7.6

Para a demonstração do *Teorema 7.2 - Equivalência entre Máquina de Turing e Máquina de Post, na simulação* da Máquina de Turing pela Máquina de Post, é necessário que a Máquina de Post realize rotação no conteúdo de X, fazendo com que o último símbolo de X passe a ser o primeiro, ou seja, se o conteúdo da variável X é $a_1\ a_2\ ...\ a_{n-1}\ a_n$ passa a ser $a_n\ a_1\ a_2\ ...\ a_{n-1}$. Desenvolva uma máquina que processe essa rotação.

Exercício 7.7

Desenvolva Autômatos com Duas Pilhas ou Máquinas com Pilhas, determinísticas ou não, que aceitem as linguagens do Exercício 7.2**.**

Exercício 7.8

Prove que uma Máquina com n Pilhas ($n > 2$) é equivalente ao uma Máquina com Duas Pilhas.

Exercício 7.9

Sobre não-determinismo:

a) O que é e quais suas principais características?

b) Qual o seu efeito, em termos de poder computacional, nas máquinas apresentadas?

Exercício 7.10

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

a) A leitura numa Máquina de Post é destrutiva, pois o símbolo lido é retirado da variável X
b) A variável X não possui tamanho finito;
c) A variável X é do tipo fila, ou seja, o último símbolo gravado é o primeiro a ser lido;
d) O símbolo # é o único símbolo auxiliar;
e) A máquina de Post pode ficar em *loop* infinito.

Exercício 7.11

Sobre a Máquina de Post, analise as seguintes afirmações:

I. A concatenação do símbolo # no fim da pilha tem de ser feita na primeira instrução do diagrama de fluxo; F
II. Um diagrama de fluxos tem sempre n+2 ramificações após uma atribuição, onde n é o numero de símbolos do alfabeto de entrada; F
III. A pilha da máquina de Post é infinita. F

Marque a alternativa correta:

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

Exercício 7.12

Sobre a Máquina de Post, marque a alternativa correta:

a) Pára quando o programa atinge uma instrução de parada (aceita ou rejeita) ou por indefinição;
b) Um programa em Post sempre contém exatamente uma instrução de partida e duas de parada (aceita e rejeita);
c) No início da execução de um programa, a fita sempre contém a palavra de entrada concatenada com o símbolo de final de fila #;
d) Após a leitura de um símbolo, o programa pode retirá-lo da fila;
e) Um programa em Post é uma seqüência finita de instruções de quatro tipos possíveis (partida, parada, desvio e atribuição).

Exercício 7.13

Sobre a Máquina de Post, analise as seguintes afirmações:

I. O programa para esta classe de máquinas é total; V
II. A Variável X é do tipo fila e é utilizada como memória de trabalho, entrada e saída; V
III. As instruções do tipo desvio podem ser definidas como funções totais.

Marque a alternativa correta: F

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

Exercício 7.14

Sobre a simulação do formalismo Máquina de Turing pelo do formalismo Máquina de Post, analise as seguintes afirmações:

I. O estado inicial é simulado pela instrução partida, seguido pela execução de um desvio, e os estados finais, simulados pela instrução aceita, sendo os demais estados simulados por instruções de desvio; V
II. A variável X é simulada pela fita e a posição da cabeça da fita representa a posição mais à esquerda da fila; V
III. Para simular o movimento para a esquerda da cabeça da fita é necessário executar tantos desvios quantos forem os símbolos da palavra corrente; V
IV. Uma indefinição na função programa da Máquina de Turing é simulada em Post pela instrução rejeita. V

Marque a alternativa correta:

a) Apenas I, II e III estão corretas;
b) Apenas II e III estão corretas;
c) Apenas I e IV estão corretas;
d) Apenas I, III e IV estão corretas;
e) Nenhuma das alternativas anteriores esta correta.

Exercício 7.15

Sobre a Família das Máquinas com Pilhas, analise as seguintes afirmações:

I. Uma Máquina Finita, que corresponde a uma Máquina Sem Pilhas, possui um poder computacional relativamente restrito, pois não tem memória auxiliar de trabalho; V

II. A Classe das Máquinas sem Pilhas é de fundamental importância no estudo das Linguagens Formais como, por exemplo, no desenvolvimento de reconhecedores de linguagens de programação tipo Pascal; F

III. A Classe das Máquinas com Uma Pilha possui poder computacional maior que a da Classe das Máquinas sem Pilhas, sendo capaz, inclusive, de reconhecer um triplo balanceamento. F

Marque a alternativa correta:

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

Exercício 7.16

Sobre a Família das Máquinas com Pilhas, marque a alternativa correta:

a) Uma máquina com pilhas não possui alfabeto auxiliar, mas pode usar símbolos do alfabeto de entrada como símbolos auxiliares;
b) Uma máquina finita (sem pilhas) determinística não pode reconhecer a linguagem duplo-balanceamento, mas uma não-determinística pode;
c) Uma máquina com uma pilha não é uma Máquina Universal, a menos que seja não-determinística;
d) Assim como na Máquina de Post, a Máquina com Pilhas possui uma fita somente para leitura
e) Nenhuma das anteriores.

Exercício 7.17

No formalismo Máquina com Pilhas, existe maneira de aumentar o poder computacional, realizando alguma das alterações abaixo? Marque a alternativa correta:

a) Múltiplas pilhas;
b) Não-Determinismo;
c) Múltiplas cabeças;
d) Maior número de instruções básicas;
e) Nenhuma das anteriores.

Exercício 7.18

Sobre Máquina com Pilhas, marque a alternativa errada (caso todas estejam corretas, marque a última alternativa):

a) X ← Ler(X) denota uma leitura, onde se lê o símbolo mais à esquerda de X ou do topo de Y, não retirando da estrutura o símbolo lido;
b) Uma máquina com pilhas é uma dupla M = (Σ, D) onde Σ é o alfabeto de símbolos de entrada e D é o diagrama de fluxos;
c) Uma máquina com pilhas reconhece o triplo balanceamento;
d) Uma máquina com pilhas possui a memória de entrada separada das memórias de trabalho e de saída;
e) Nenhuma das anteriores.

Exercício 7.19

Sobre Autômato com Duas Pilhas, marque a alternativa errada:

a) Um Autômato com Duas Pilhas é composto basicamente por: Fita, Pilhas, Unidade de Controle e Função de Transição;
b) A situação inicial de cada pilha é vazia;
c) Uma importante diferença entre um Autômato com Duas Pilhas e uma Máquina com Duas Pilhas é que no autômato o programa é especificado usando a noção de estados e transições e na máquina é especificado como um fluoxograma.
d) O que reflete o estado corrente da máquina é a Unidade de Controle.
e) A função programa ou função de transição pode não ser total.

Exercício 7.20

Sobre Autômato com Duas Pilhas $M = (\Sigma, Q, \Pi, q_0, F, V)$, marque a alternativa errada:

a) Σ indica o alfabeto de entrada;
b) Q indica o conjunto de estados possíveis;
c) Π indica a função de transição;
d) F indica que a palavra é falsa;
e) V indica o alfabeto auxiliar.

Exercício 7.21

Sobre a família dos Autômato com Pilhas, analise as seguintes afirmações:

I. Para cada Autômato Não-Determinístico, sempre há um Autômato Determinístico com Duas Pilhas equivalente. V
II. Pode-se simular um Autômato Finito por uma Máquina de Turing e vice-versa. F
III. As estruturas de dados do tipo pilha podem ser usadas livremente para leitura, gravação e entrada de dados. F
IV. Pode ocorrer de um Autômato, para uma dada palavra de entrada, nunca atingir uma condição de parada. V

Marque a alternativa correta:

a) Apenas I e II estão corretas;
b) Apenas I, II e III estão corretas;
c) I é correta e III é errada;
d) III e IV são erradas;
e) I, II, III e IV estão corretas.

Exercício 7.22

Relacionando a Hierarquia das Classes de Máquinas com a Hierarquia das Classes de Linguagens, numere a primeira coluna de acordo com a segunda:

( ) Linguagens Recursivas	1. Máquinas Determinísticas com 1 pilha
( ) Linguagens Regulares	2. Máquinas Universais
( ) Linguagens Livres do Contexto	3. Máquinas Determinísticas ou Não-Determinísticas sem pilha
( ) Linguagens Enumeráveis Recursivamente	4. Máquinas Universais – Algoritmos que sempre param
( ) Linguagens Livres do Contexto Determinísticas	5. Máquinas Não-determinísticas com 1 pilha

A numeração correta, de cima para baixo, é:

a) 2 – 5 – 4 – 1 – 3
b) 4 – 5 – 1 – 2 – 3
c) 4 – 5 – 2 – 3 – 1
d) 4 – 3 – 5 – 2 – 1
e) 2 – 4 – 5 – 1 – 3

Exercício 7.23

Sobre não-determinismo, analise as seguintes afirmações:

I. Uma máquina de Turing não-determinística possui um poder computacional superior ao de uma máquina de Post determinística; F
II. Uma máquina de Turing possui poder computacional superior ao de um máquina finita, sendo assim capaz de simulá-la; V
III. Está demonstrado que nenhuma outra forma de expressar algoritmos terá capacidade superior à da máquina de Turing.

Marque a alternativa correta:

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

Exercício 7.24

Sobre não-determinismo, analise as seguintes afirmações:

I. Somente Máquinas com uma pilha não- determinística tem a capacidade para reconhecer a linguagem:

$$Palavra\_Reversa = \{ ww^r \mid w \text{ pertence a } \{ a, b \}^* \}$$ F

II. O Não-Determinismo pode modificar a eficiência computacional das Máquinas Universais estudadas; F

III. Uma palavra w pertencente a uma linguagem L é entrada em uma máquina M não-determinística. Sabendo que em um caminho alternativo a palavra foi aceita, em outro foi rejeitada, e em todos os demais ficou em *loop*, então pode-se afirmar que w pertence a $LOOP(M)$. F

Marque a alternativa correta:

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