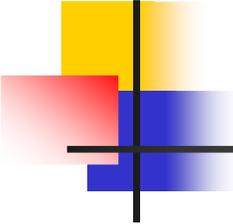


# Teoria da Computação

---

## Unidade 3 – Máquinas Universais (cont.)

Referência – Teoria da Computação (Divério, 2000)



# Máquina com Pilhas

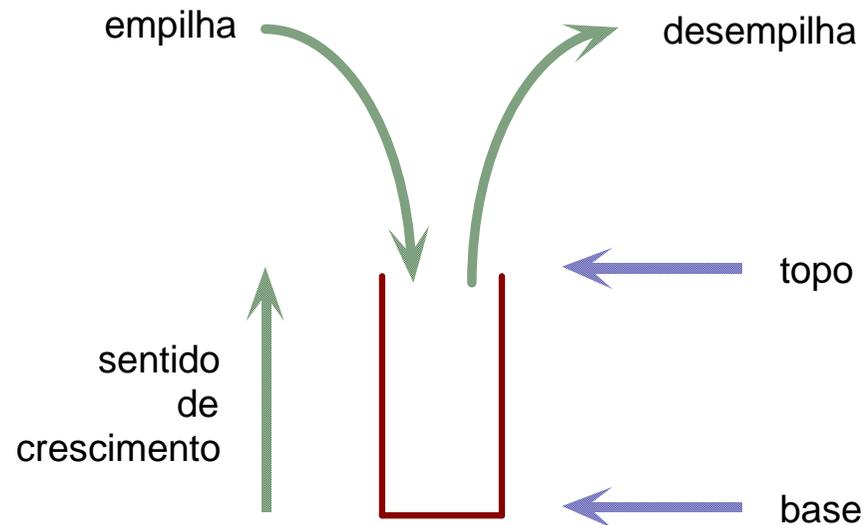
---

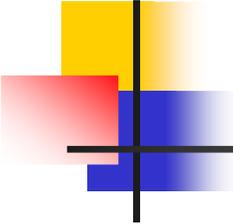
- Diferencia-se das MT e MP pelo fato de possuir a memória de entrada separada das memórias de trabalho e saída.
- Memória auxiliar:
  - tipo pilha;
  - cada máquina possui zero ou mais pilhas;
  - as pilhas não tem limitação de tamanho.
- Duas ou mais pilhas: mesmo poder computacional que a Classe das MT ou MP → Linguagens recursivamente enumeráveis

# Outros Modelos de Máquinas Universais

## ■ Máquina com Pilhas

- Formalizada por vários autores na década de 60

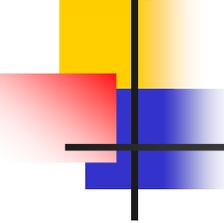




## Máquina com Pilhas

---

- Possui um programa associado (fluxograma)
  - Partida;
  - Parada;
  - Desvio condicional (desempilha);
  - Empilha.



## Máquina com Pilhas - definição

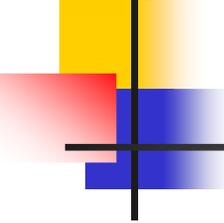
---

- Uma máquina com pilhas é uma dupla

$$M = (\Sigma, D)$$

$\Sigma$  *alfabeto de símbolos de entrada;*

$D$  *programa* ou *diagrama de fluxos* construído a partir dos componentes elementares: *partida, parada, desvio, empilha e desempilha.*



## Máquina com Pilhas - definição

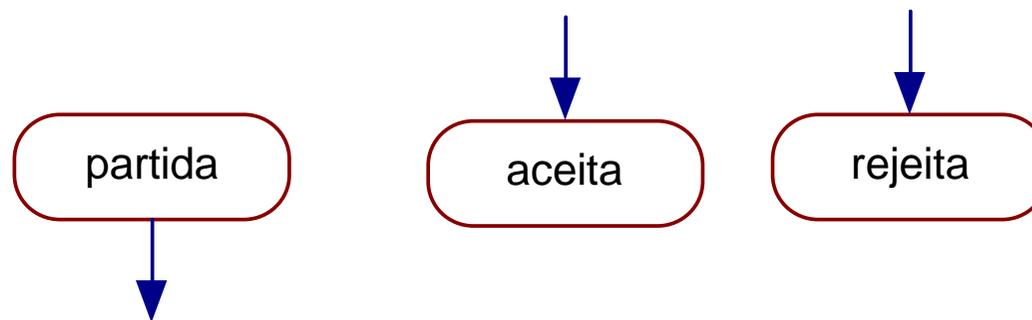
---

- Consiste basicamente de três partes
  - a) **Variável X**: representa a fita entrada;
  - b) **Variáveis  $y_i$** : ( $i \geq 0$ ) Do tipo **Pilha**, utilizadas como memória de trabalho;
  - c) **Programa**. É uma sequência finita de **instruções**, representado como um diagrama de fluxos onde cada vértice é uma instrução.

# Máquina com Pilhas - definição

## ■ Componentes

- 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**).



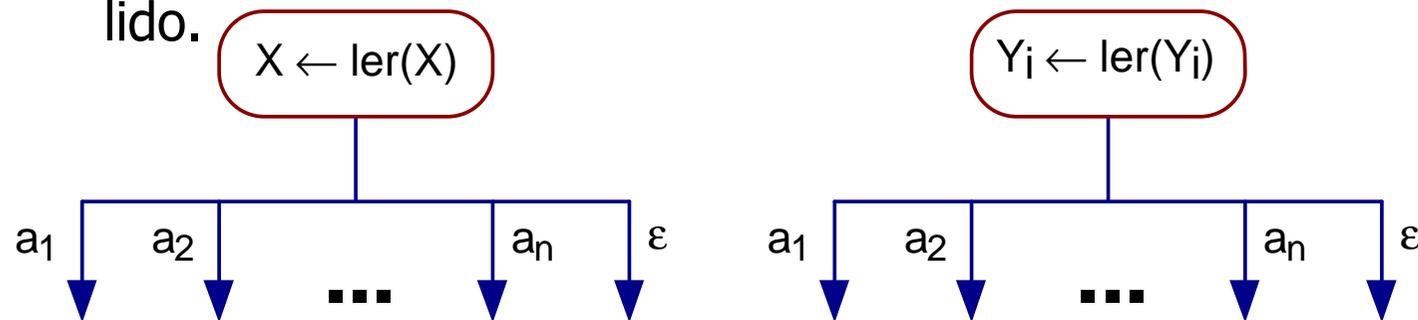
# Máquina com Pilhas

## Componentes (cont.)

- c) *Desvio* (ou *Teste* em X) e *Desempilha* (em Y): 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  $\Sigma$  é  $n$ , então existem  $n+1$  arestas de desvios condicionais, pois se deve incluir a possibilidade da palavra vazia  $\epsilon$ .

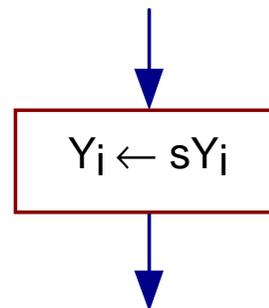
$X \leftarrow \text{ler}(X)$  denota uma *leitura destrutiva*, que lê o símbolo mais à esquerda de X ou do topo de  $Y_i$ , retirando o símbolo lido.

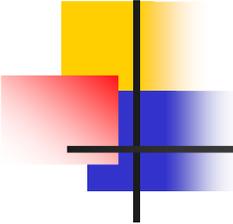


# Máquina com Pilhas

## ■ Componentes (cont.)

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

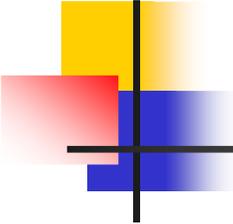




# Máquina com Pilhas

---

- Diagrama de Fluxos
  - Existe somente uma partida, mas podem existir diversas (zero ou mais) instruções de parada (aceitação / rejeição) ou ficar em *loop infinito*
  - Em um desvio (e **desempilha**), se  $X$  (e  $Y_i$ ) contém  $\varepsilon$ , então segue o fluxo correspondente. Caso contrário, lê o símbolo mais à esquerda de  $X$  (no **topo de  $Y_i$** ) e remove-o após a decisão da próxima instrução



## Máquina com Pilhas

---

- Exemplo – Duplo Balanceamento

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

- A Máquina com Pilhas:

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

- onde  $D$  é, ...(fluxograma)..., tal que:

$$\text{ACEITA}(\text{Pilhas\_Duplo\_Bal}) = \text{Duplo\_Bal}$$

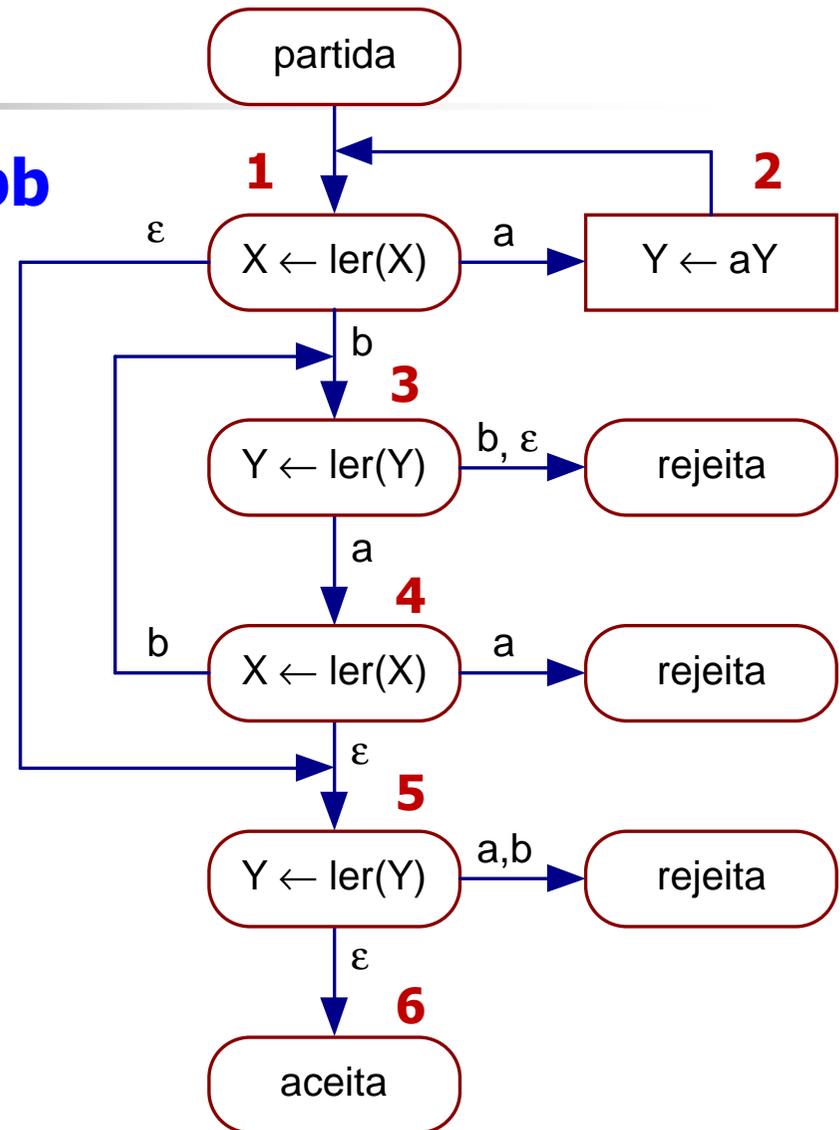
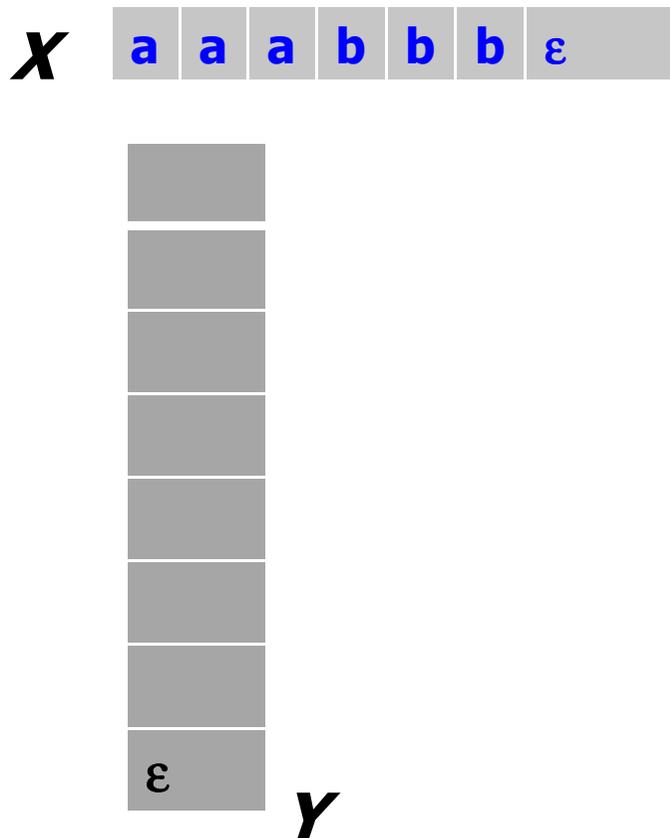
$$\text{REJEITA}(\text{Pilhas\_Duplo\_Bal}) = \Sigma^* - \text{Duplo\_Bal}$$

$$\text{LOOP}(\text{Pilhas\_Duplo\_Bal}) = \emptyset.$$



# Máquinas com Pilhas

- Simulando a entrada **aaabbb**



# Máquinas com Pilhas

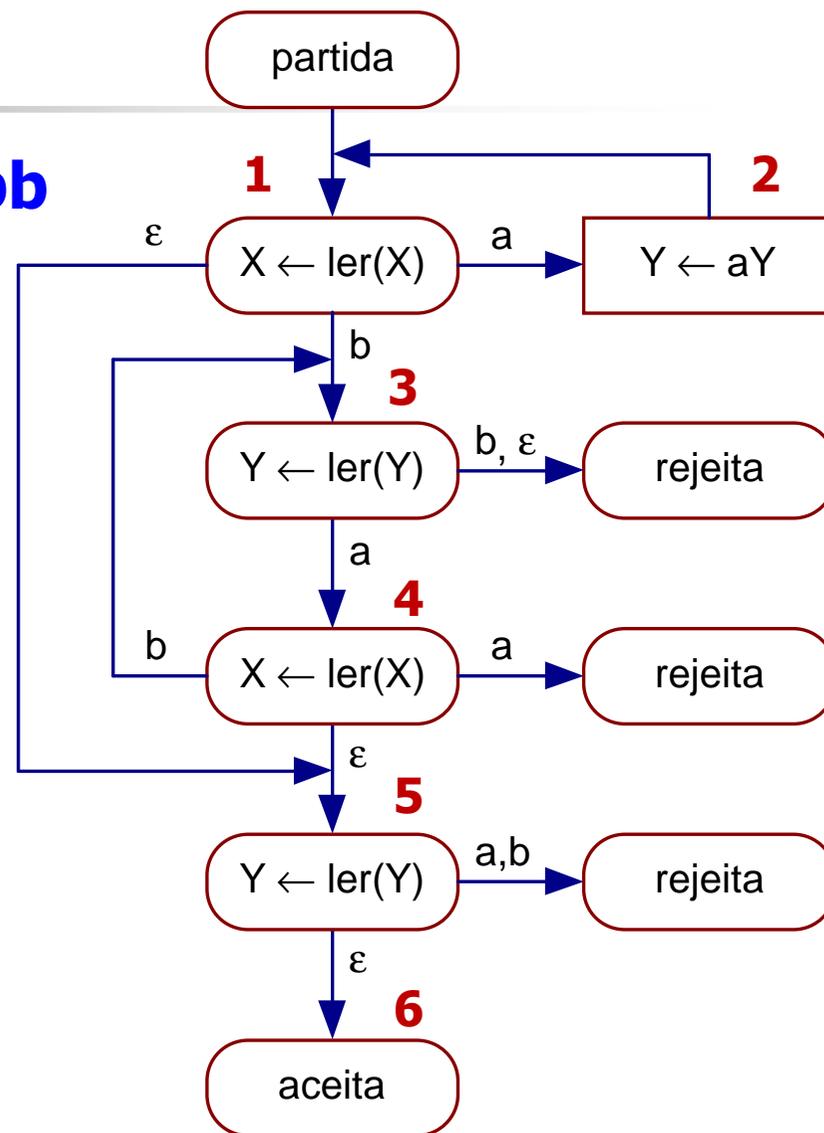
- Simulando a entrada **aaabbb**

**X** a a b b b  $\epsilon$



**Passos 1;2**

**Y**



# Máquinas com Pilhas

- Simulando a entrada **aaabbb**

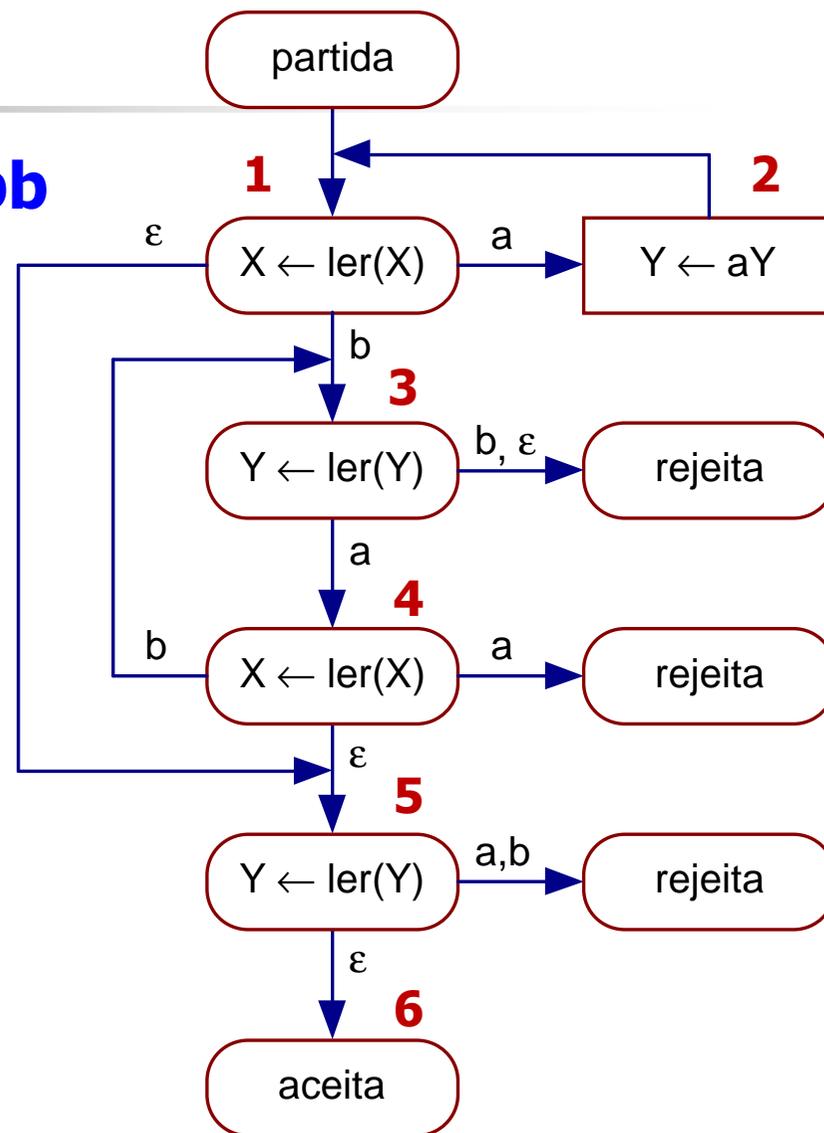
**X** a b b b  $\epsilon$

a  
 a  
 $\epsilon$

**Passos 1;2**

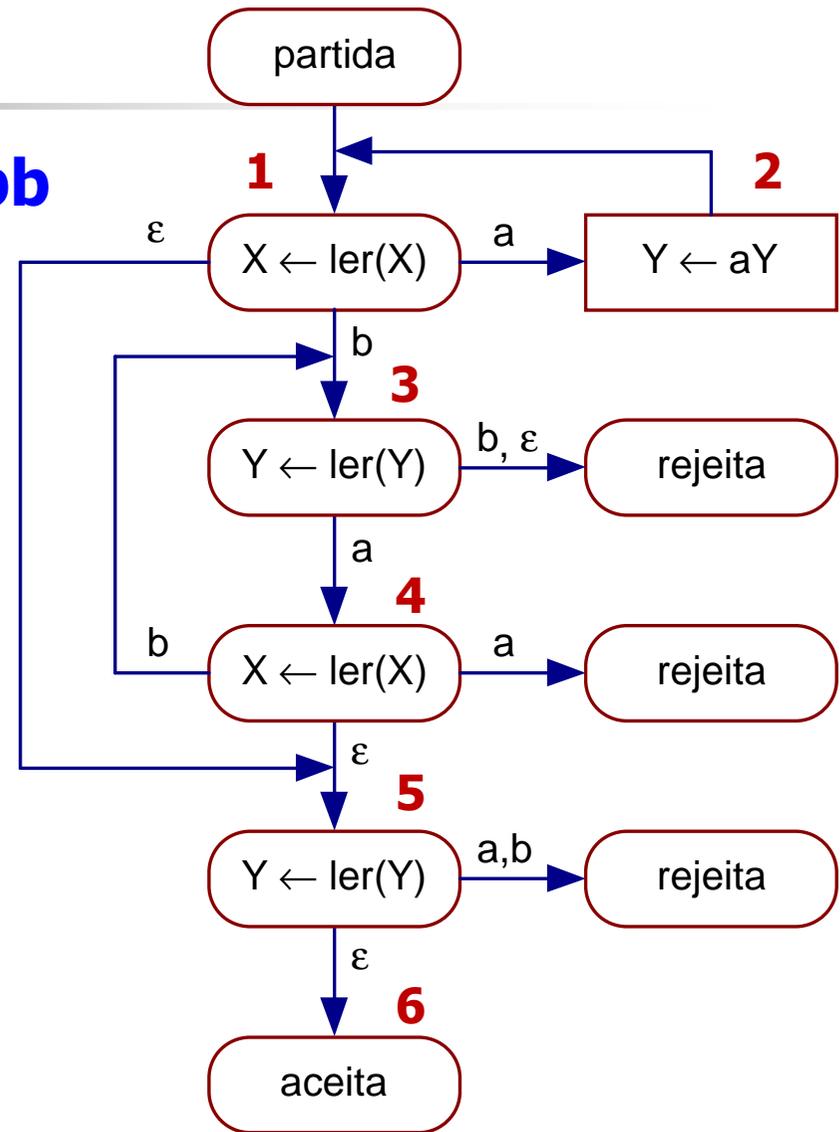
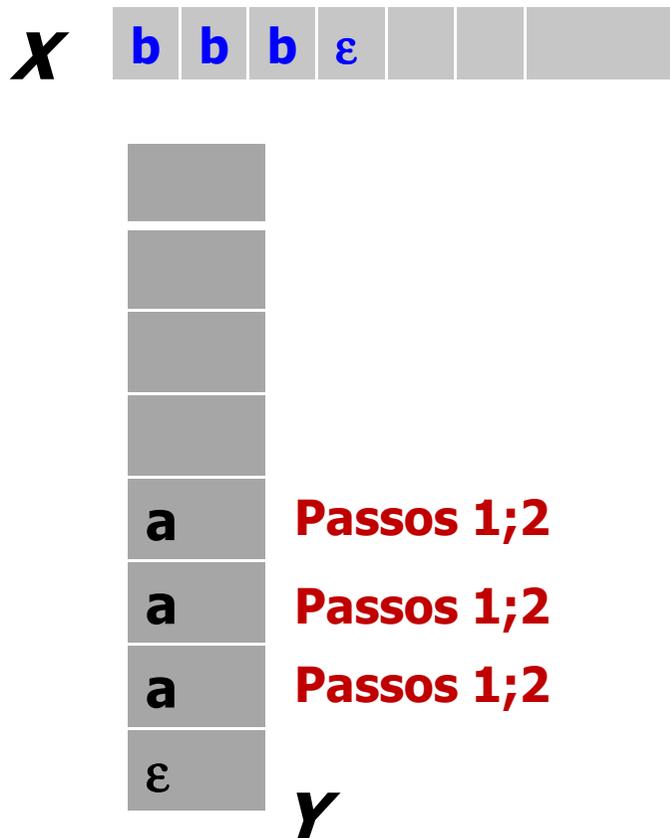
**Passos 1;2**

**Y**



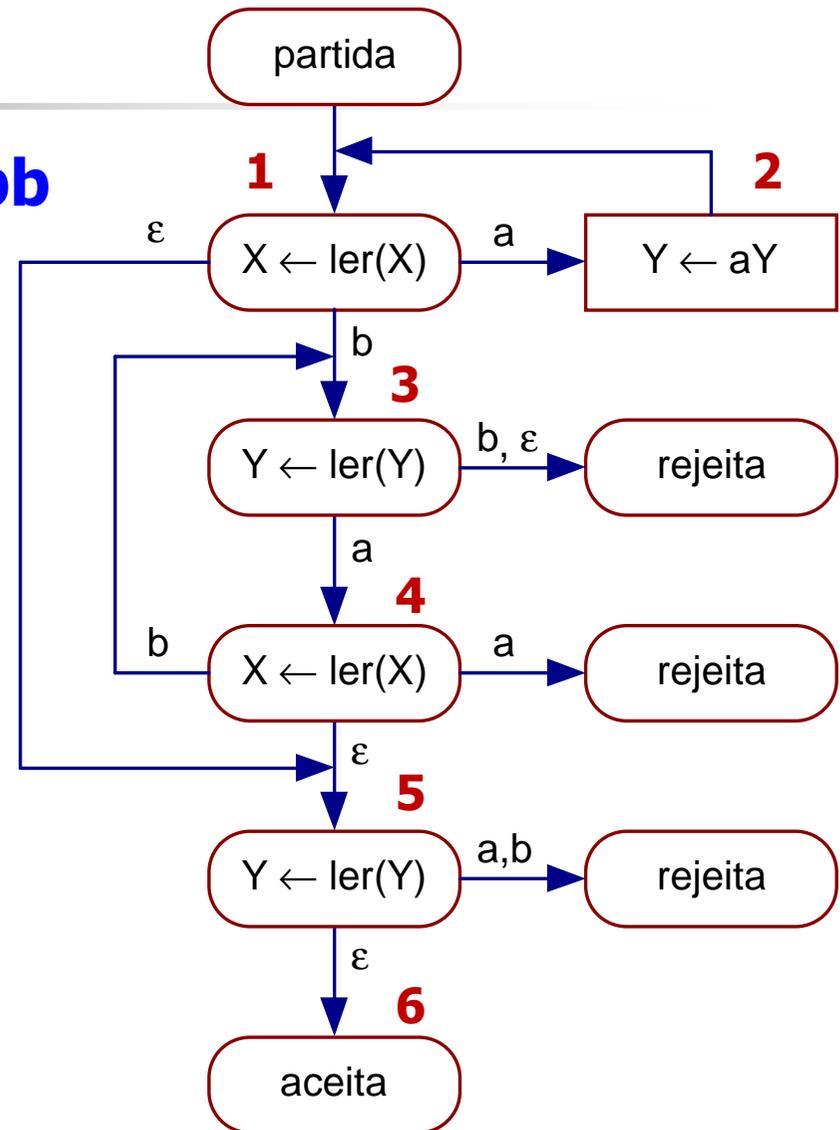
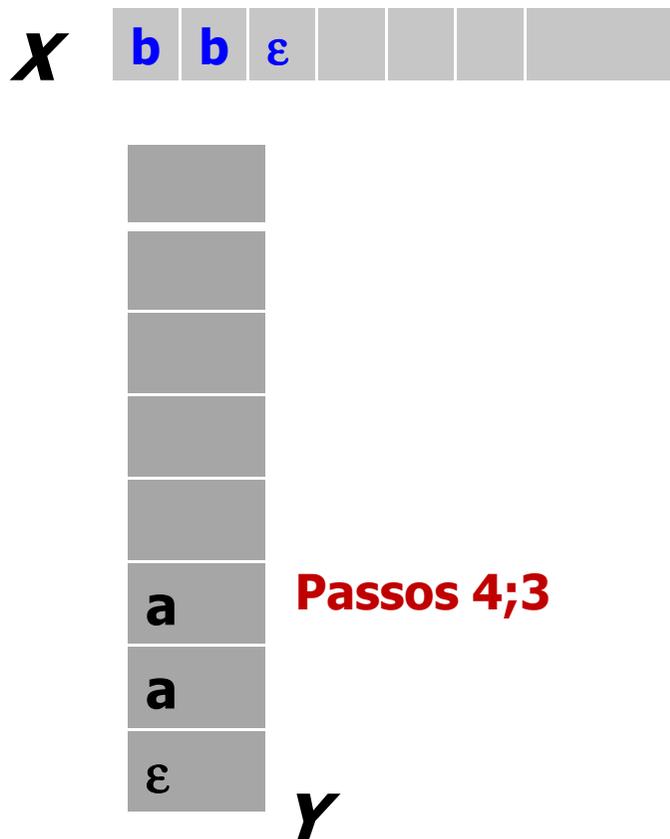
# Máquinas com Pilhas

- Simulando a entrada **aaabbb**



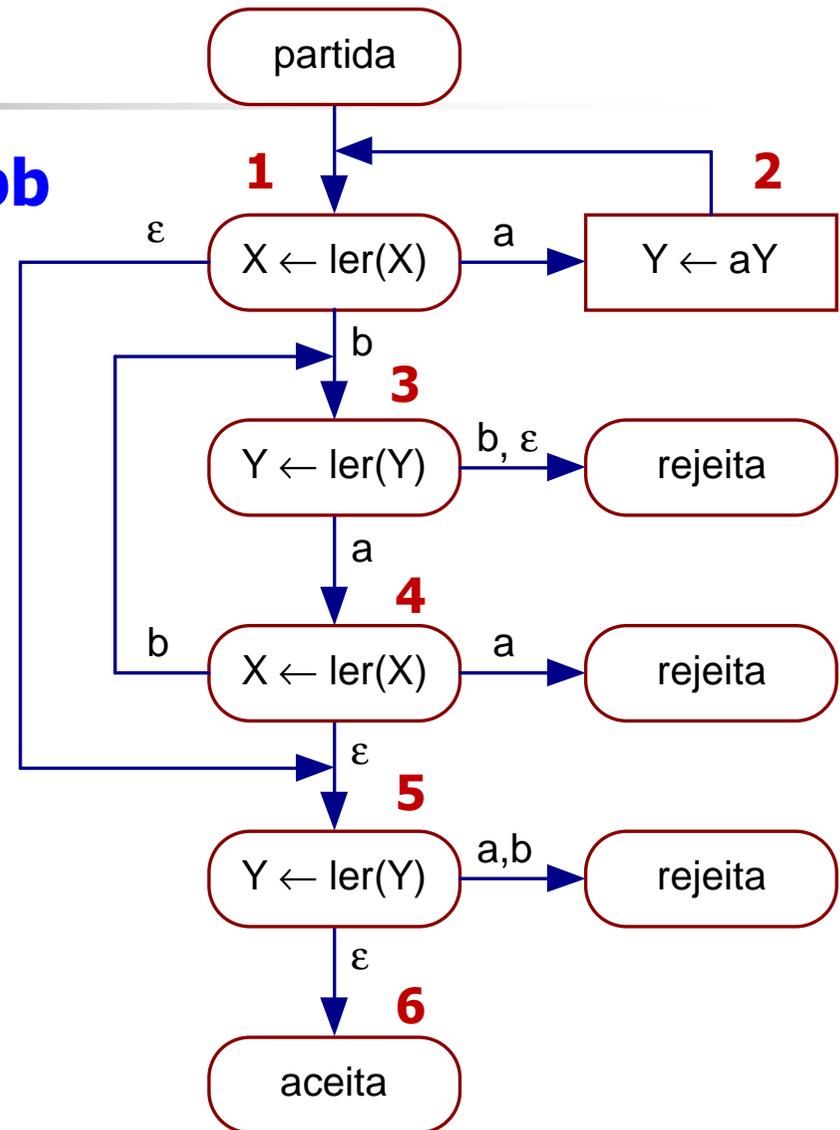
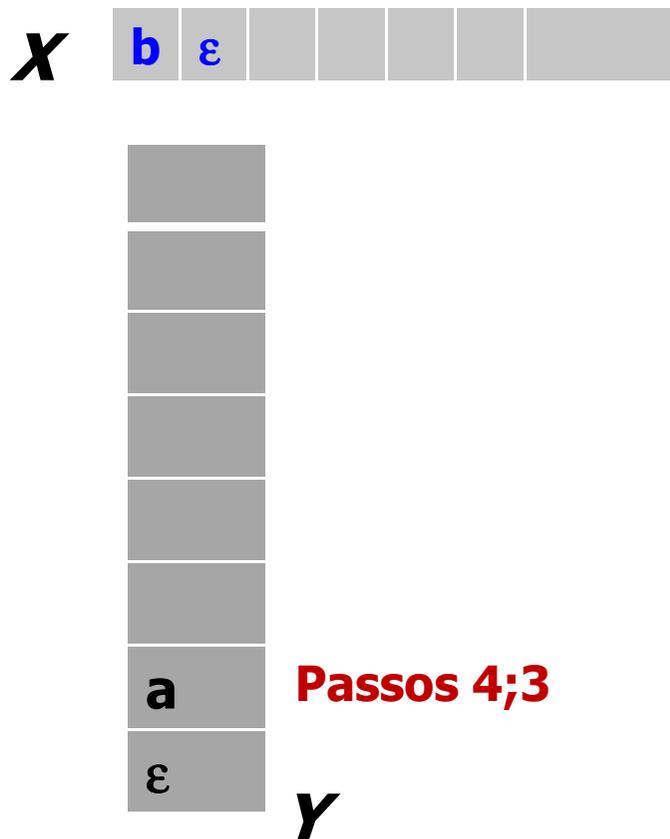
# Máquinas com Pilhas

- Simulando a entrada **aaabbb**

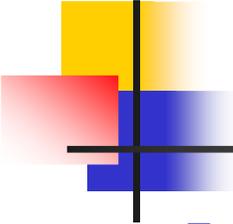


# Máquinas com Pilhas

- Simulando a entrada **aaabbb**







# Máquinas com Pilhas

---

- *Exemplo – Prefixo*
- Linguagem:

$\text{Prefixo\_aaa} = \{w \mid w \in \{a,b\}^* \text{ e } w \text{ contém a subpalavra } aaa \text{ como prefixo}\}$

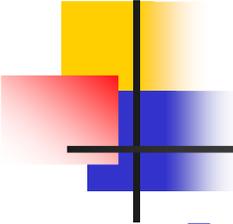
Ex: aaabab, aaaaaab e aaa são palavras de Prefixo\_aaa

$\text{Pilhas\_Prefixo\_aaa} = (\{a, b\}, D)$  onde  $D$  é tal que:

$\text{ACEITA}(\text{Pilhas\_Prefixo\_aaa}) = \text{Prefixo\_aaa}$

$\text{REJEITA}(\text{Pilhas\_Prefixo\_aaa}) = \Sigma^* - \text{Prefixo\_aaa}$

$\text{LOOP}(\text{Pilhas\_Prefixo\_aaa}) = \emptyset$



# Máquinas com Pilhas

---

- *Exemplo – Prefixo*
- Linguagem:

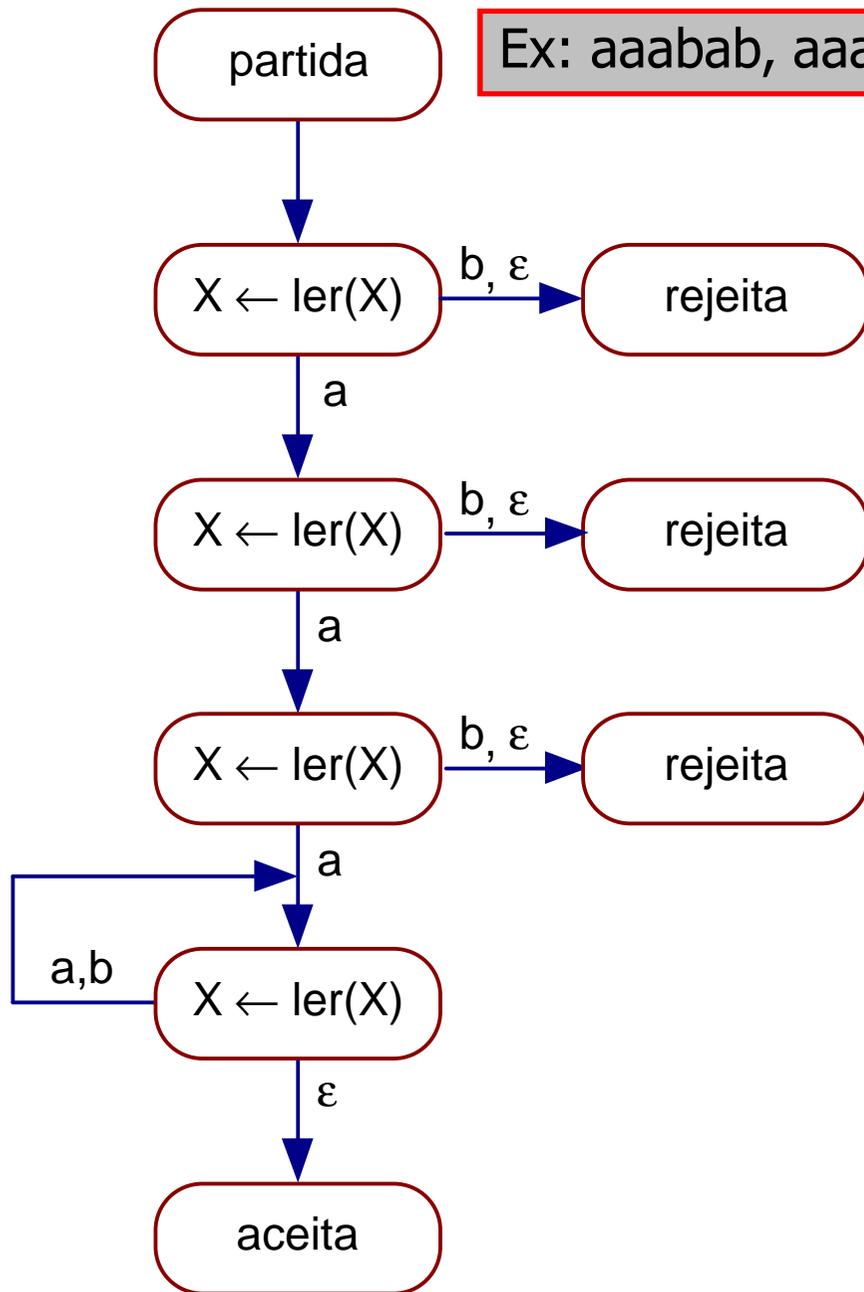
$\text{Prefixo\_aaa} = \{w \mid w \in \{a,b\}^* \text{ e } w \text{ contém a subpalavra } aaa \text{ como prefixo}\}$

Ex: aaabab, aaaaaab e aaa são palavras de Prefixo\_aaa

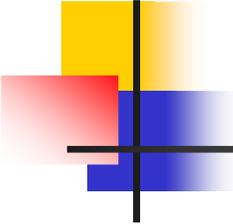
$\text{Pilhas\_Prefixo\_aaa} = (\{a, b\}, D)$  onde  $D$  é tal que:

Quantas pilhas são necessárias  
para o reconhecimento?

Ex: aaabab, aaaaaab e aaa são palavras de Prefixo\_aaa



- Não foram utilizadas pilhas. O algoritmo lê o prefixo **aaa** executando três leituras. Após, qualquer sufixo é aceito. Tal linguagem pode ser reconhecido utilizando um autômato finito



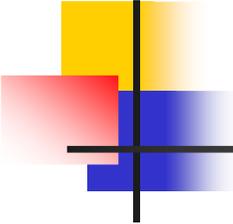
# Máquina com Pilhas

---

- Triplo Balanceamento

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

Quantas pilhas são necessárias para o reconhecimento?



# Máquina com Pilhas

---

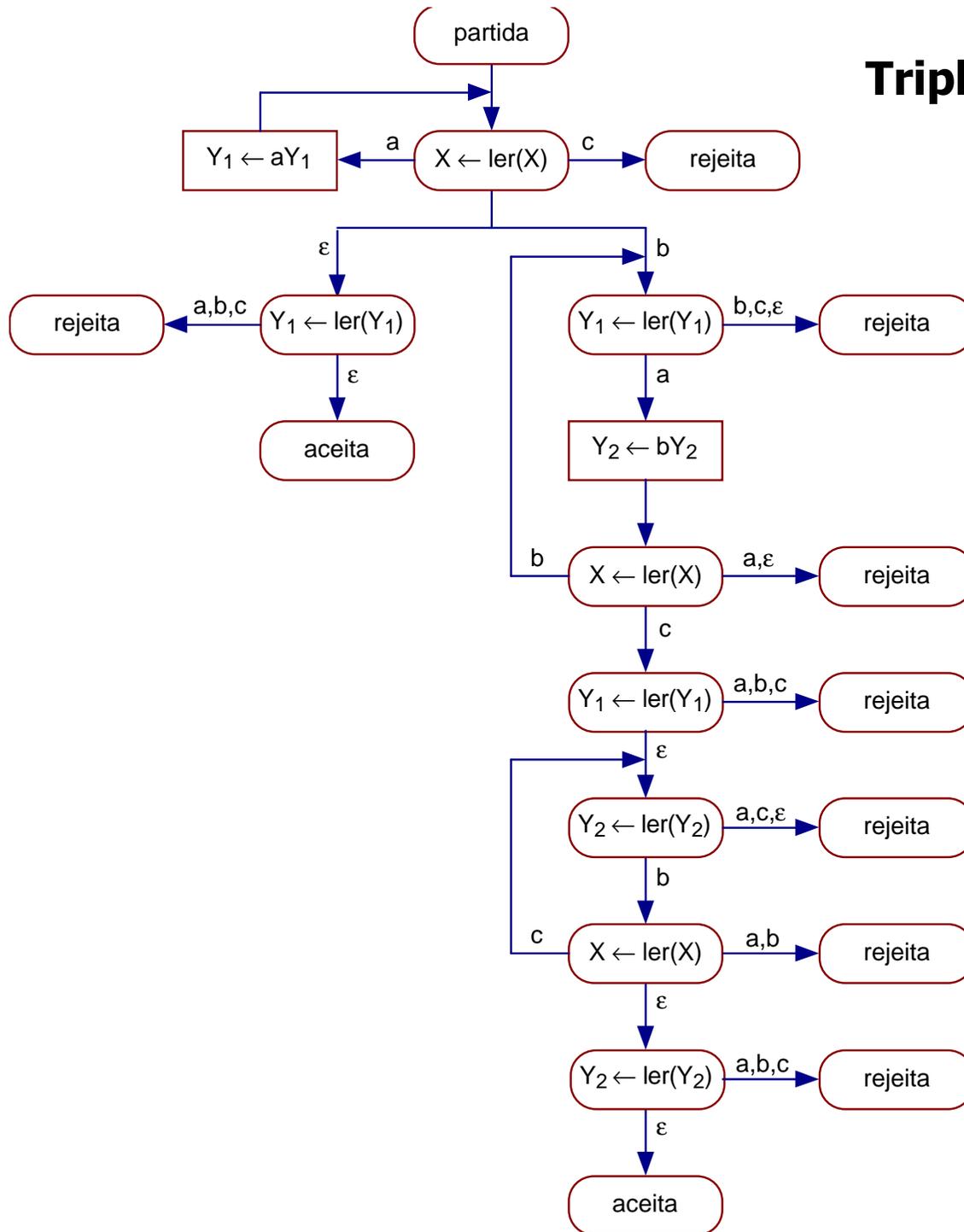
- Triplo Balanceamento

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

- Estratégia: usar duas pilhas Y1 e Y2

- Inserir os símbolos **a** em Y1;
- Para cada símbolo **b** remover **a** em Y1 e inserir **b** em Y2 → garantindo assim que a quantidade de a's e b's são idênticas;
- Para cada símbolo **c** remover **b** em Y2 → garantindo assim que a quantidade de b's e c's são idênticas.

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



Estratégia: usar duas pilhas Y1 e Y2

- Inserir os símbolos **a** em Y1;
- Para cada símbolo **b** remover **a** em Y1 e inserir **b** em Y2 → garantindo assim que a quantidade de a's e b's são idênticas;
- Para cada símbolo **c** remover **b** em Y2 e → garantindo assim que a quantidade de a's e b's são idênticas.

# Máquina com Pilhas

- A classe de linguagens representadas por máquinas de pilhas depende de quantidade de pilhas que ela possui:
  - **Nenhuma pilha:** corresponde ao **autômato finito**, capaz de reconhecer a classe das **linguagens regulares**;

Produções do tipo

$A \rightarrow aB$  ou  $A \rightarrow a$   
 $A \rightarrow Ba$  ou  $A \rightarrow a$   
onde  $A, B \in$   
 $V_n$  (não terminais) e  
 $a \in V_t$  (terminais)

# Máquina com Pilhas

- A classe de linguagens representadas por máquinas de pilhas depende de quantidade de pilhas que ela possui:
  - Uma pilha: corresponde ao **autômato de pilha**, capaz de reconhecer a classe das **linguagens livre de contexto**;

Produções do tipo

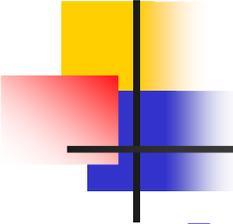
$A \rightarrow \alpha$  onde

$A \in V_n$

$\alpha \in (V_n \cup V_t)^*$

lado esquerdo da regra há apenas um símbolo não-terminal

Não é possível reconhecer a linguagem  
 $\text{Triplo\_Bal} = \{ a^n b^n c^n \mid n \geq 0 \}$



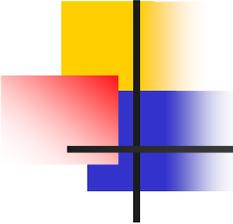
# Máquina com Pilhas

- A classe de linguagens representadas por máquinas de pilhas depende de quantidade de pilhas que ela possui:
  - **Duas pilhas:** corresponde à máquina de Turing, capaz de reconhecer a classe das linguagens recursivamente enumeráveis;

Produções do tipo

$$\alpha \rightarrow \beta \quad \alpha \in \Sigma^+; \beta \in \Sigma^*$$

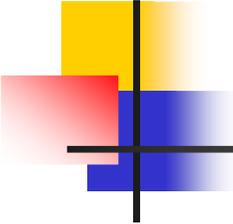
- **Três ou mais pilhas:** podem ser simuladas por uma máquina com apenas duas pilhas.



# Exercícios

---

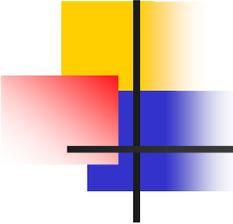
- $L1 = \{w \mid w \text{ tem o mesmo número de símbolos "a" e "b"}\}$
- $L2 = \{w \mid \text{o décimo símbolo da direita para a esquerda é "a"}\}$
- $L3 = \{waw \mid w \text{ é palavra de } \{b,c\}^*\}$



# Outros modelos de máquinas universais

---

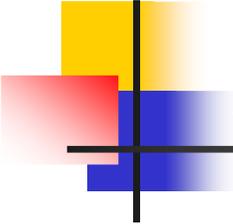
## Autômato com duas pilhas



## Autômato com Duas Pilhas

---

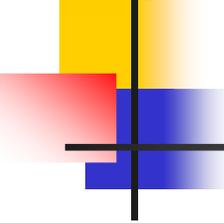
- O autômato com duas pilhas é uma máquina universal **similar à máquina com duas pilhas**. A principal diferença é que o programa é especificado utilizando a noção de estados, e não como um diagrama de fluxos. Tem o **mesmo poder computacional da MT**.
  - Diagramas de fluxo são úteis desenvolvimento de algoritmos e na visualização da estruturação e computação.
  - Máquinas com estados são mais indicadas para estudos teóricos-formais pois facilitam provas e estudos de facilidades especiais como não-determinismo, bem como são mais fáceis de serem implementados.



# Autômato com Duas Pilhas

---

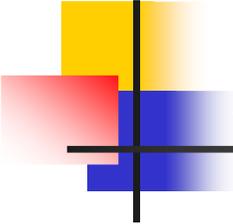
- O autômato com duas pilhas é composto, basicamente, por quatro componentes
  - a) **Fita:** dispositivo de entrada que contém a informação a ser processada;
  - b) **Duas pilhas:** memórias auxiliares que podem ser usadas livremente para leitura e gravação. Uma pilha é dividida em células, armazenando, 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.



## Autômato com Duas Pilhas

---

- c) **Unidade de Controle:** reflete o estado corrente da máquina. Possui um número finito e predefinido de estados, 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 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 a qual 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.**



# Autômato com Duas Pilhas

---

## d) Função Programa

- 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  $\epsilon$  na leitura da fita ou de alguma pilha indica que o **autômato não lê nem move a cabeça**. Pelo menos uma leitura deve ser realizada ou sobre a fita ou sobre alguma pilha;
- o símbolo  $\epsilon$  na gravação indica que **nenhuma gravação é realizada** na pilha (e não move a cabeça).

# Autômato com Duas Pilhas

- A função programa

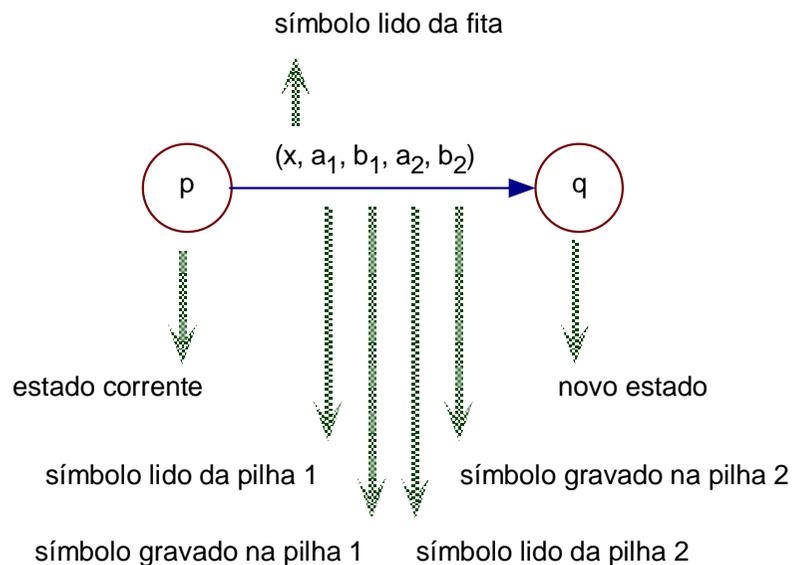
CONSIDERA	DETERMINA
Estado corrente	Novo estado
símbolo lido da fita (pode ser omitido) ou teste se toda a palavra de entrada foi lida	símbolo gravado em cada pilha (pode ser omitido)
símbolo lido de cada pilha (pode ser omitido) ou teste de pilha vazia	

$$\Pi(p, ?, a, \varepsilon) = \{ (q, \varepsilon, b) \}$$

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

# Autômato com Duas Pilhas

## Representação da função programa como um grafo



Ex: (a, ε, B, ?, ε) → Toda palavra foi lida ou pilha vazia

Leu a da fita

Grava B na Pilha 1

Não lê nem move na Pilha 1

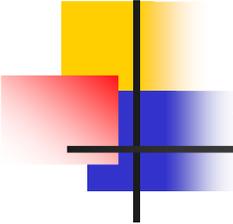
*processamento:* consiste na sucessiva aplicação  $\Pi$  para cada símbolo de  $w$  (da esquerda  $\rightarrow$  direita) até ocorrer uma condição de parada.

*ciclo infinito:* um programa que empilha e desempilha um mesmo símbolo indefinidamente, sem ler da fita.

*Condições de parada:*

*Estado Final:* O autômato assume um estado final: o autômato pára, e a palavra de entrada é aceita

*Função Indefinida:* o autômato pára, e a palavra de entrada é rejeitada.



# Autômato com Duas Pilhas

---

- Exemplo – Duplo Balanceamento
- Linguagem

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

- O Autômato com Pilhas

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

- $\Pi$ :

$$\Pi(q_0, a, \varepsilon, \varepsilon) = (q_0, B, \varepsilon)$$

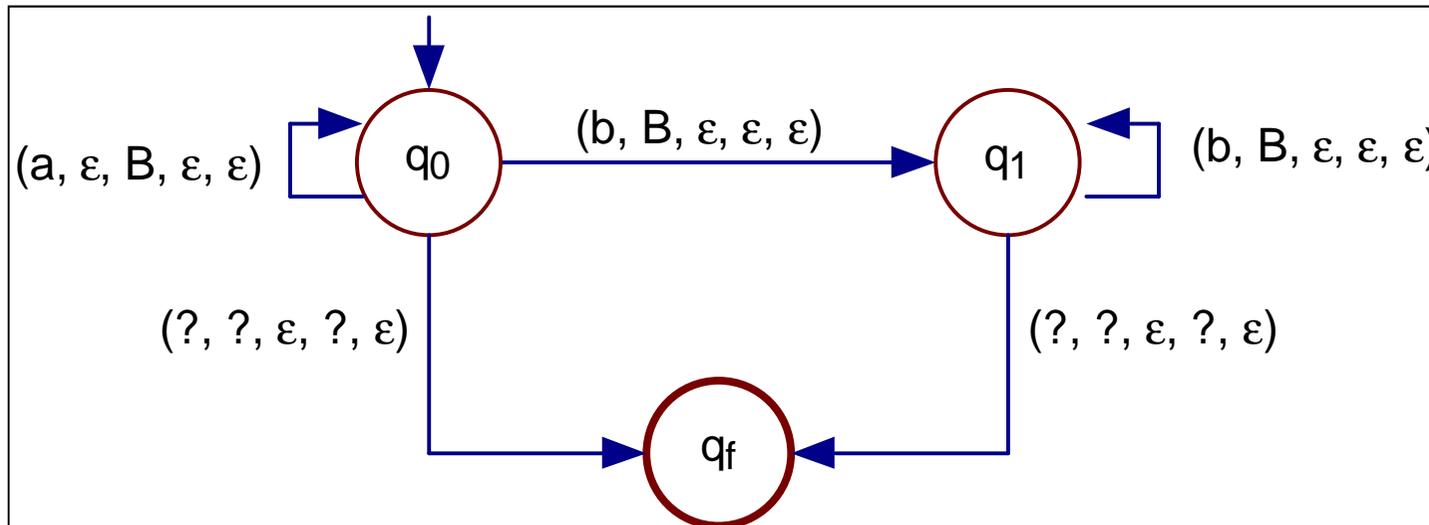
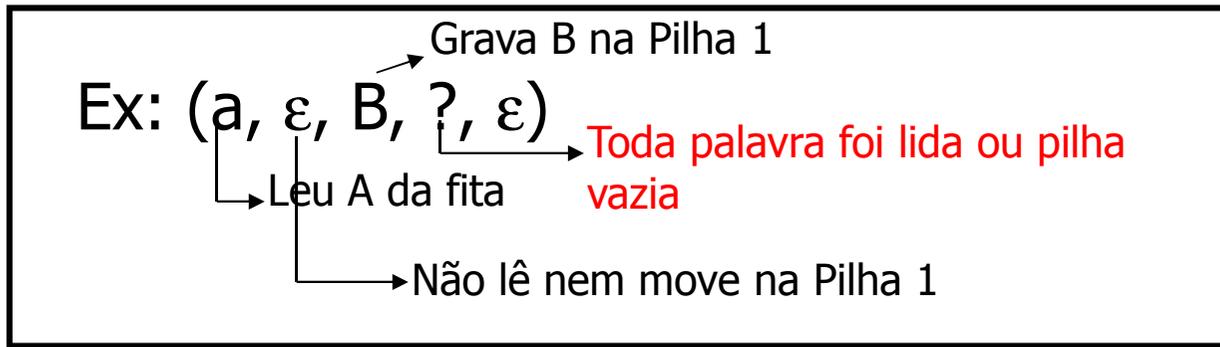
$$\Pi(q_0, b, B, \varepsilon) = (q_1, \varepsilon, \varepsilon)$$

$$\Pi(q_0, ?, ?, ?) = (q_f, \varepsilon, \varepsilon)$$

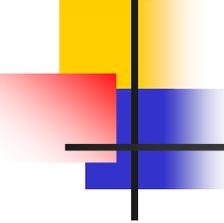
$$\Pi(q_1, b, B, \varepsilon) = (q_1, \varepsilon, \varepsilon)$$

$$\Pi(q_1, ?, ?, ?) = (q_f, \varepsilon, \varepsilon)$$

$$\text{ACEITA}(\text{A2P\_Duplo\_Bal}) = \text{Duplo\_Bal}$$



- 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  $1$ .
- O algoritmo somente aceita se, ao terminar de ler toda a palavra de entrada, as pilhas estiverem vazias.



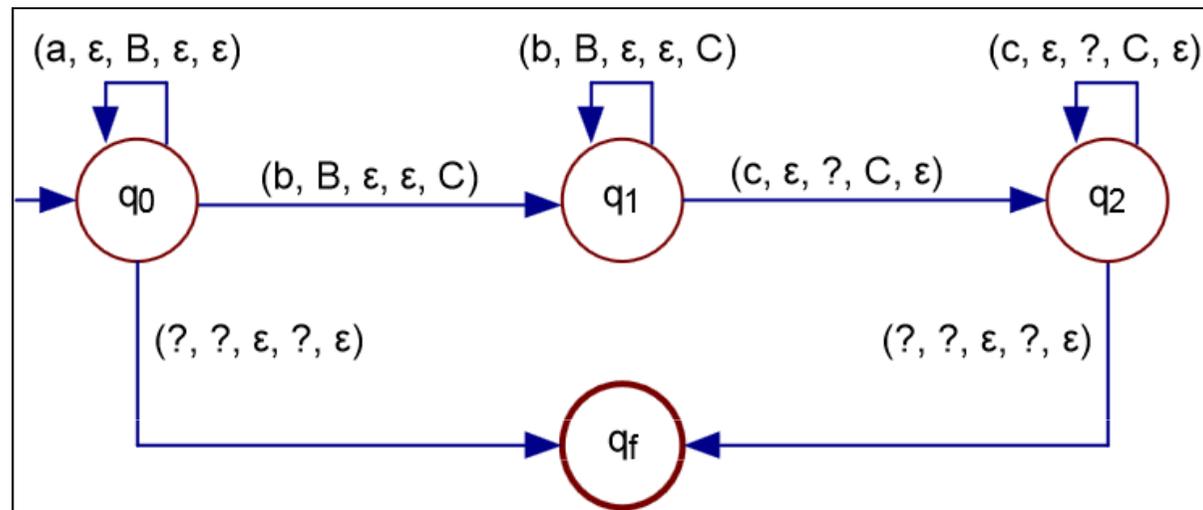
# Autômato com Duas Pilhas

---

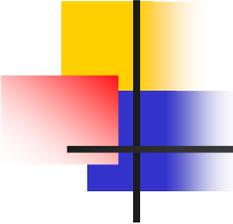
- Exercício – Triplo Balanceamento
- Linguagem

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

# Autômato com Duas Pilhas



- 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 um símbolo  $C$  na pilha 2.
- 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.

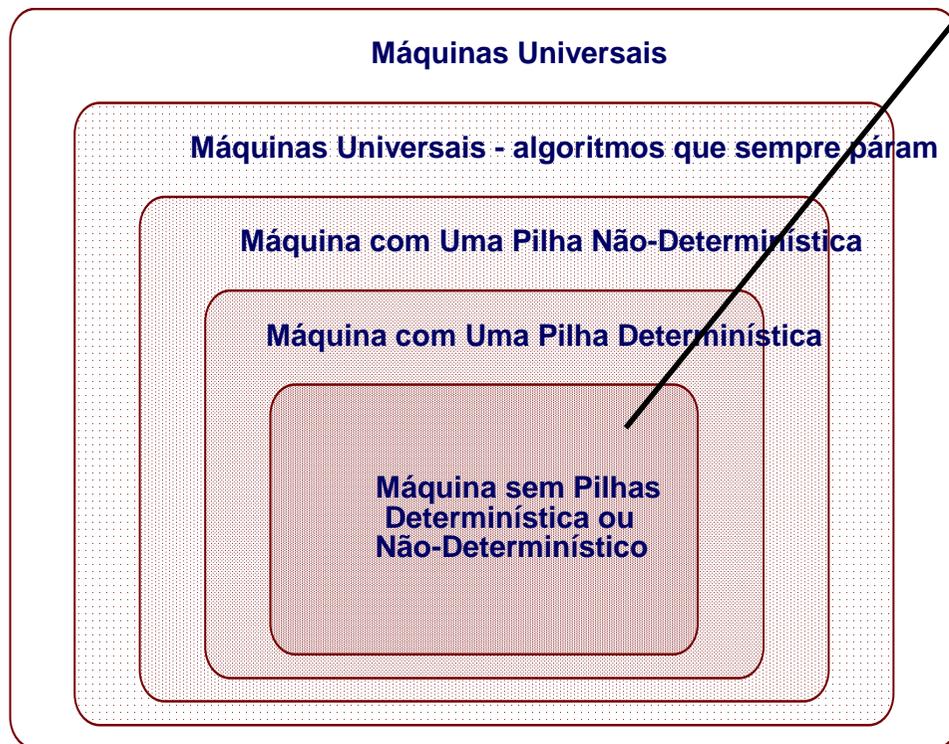


# Máquinas com Pilhas

---

*Conclusões sobre o número de pilhas e o poder computacional das máquinas com pilhas*

# Hierarquia de Classes de Máquinas



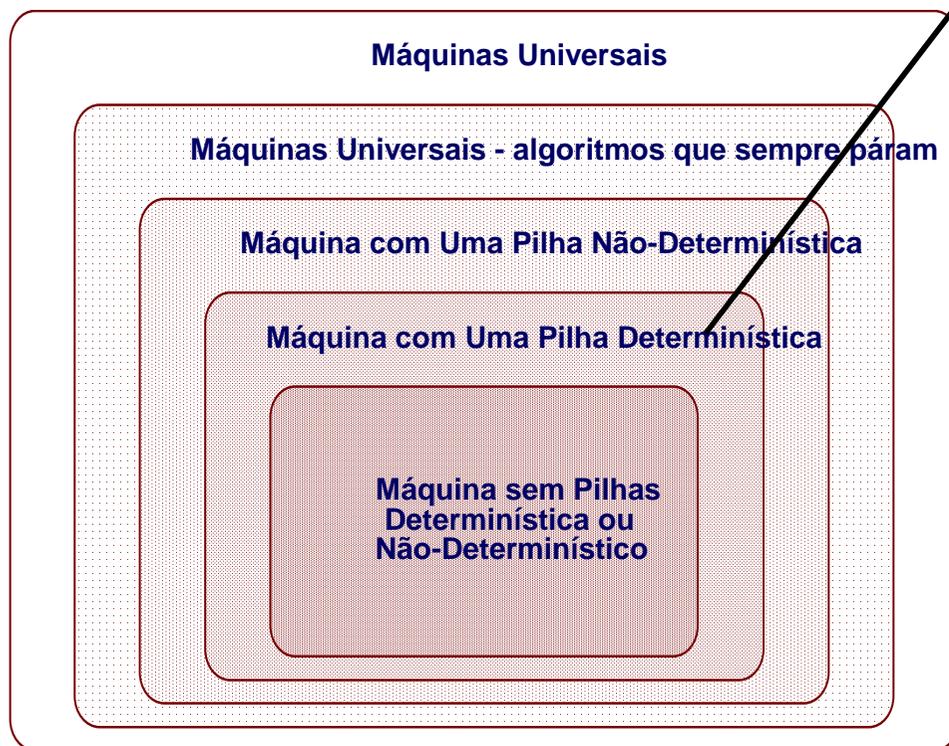
*Linguagens Regulares* Correspondem à Classe das Máquinas sem Pilha.

- 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.

## Formalismos

Gramática e expressão regular  
Autômato finito

# Hierarquia de Classes de Máquinas



## *Linguagens Livres do Contexto Determinísticas*

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

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

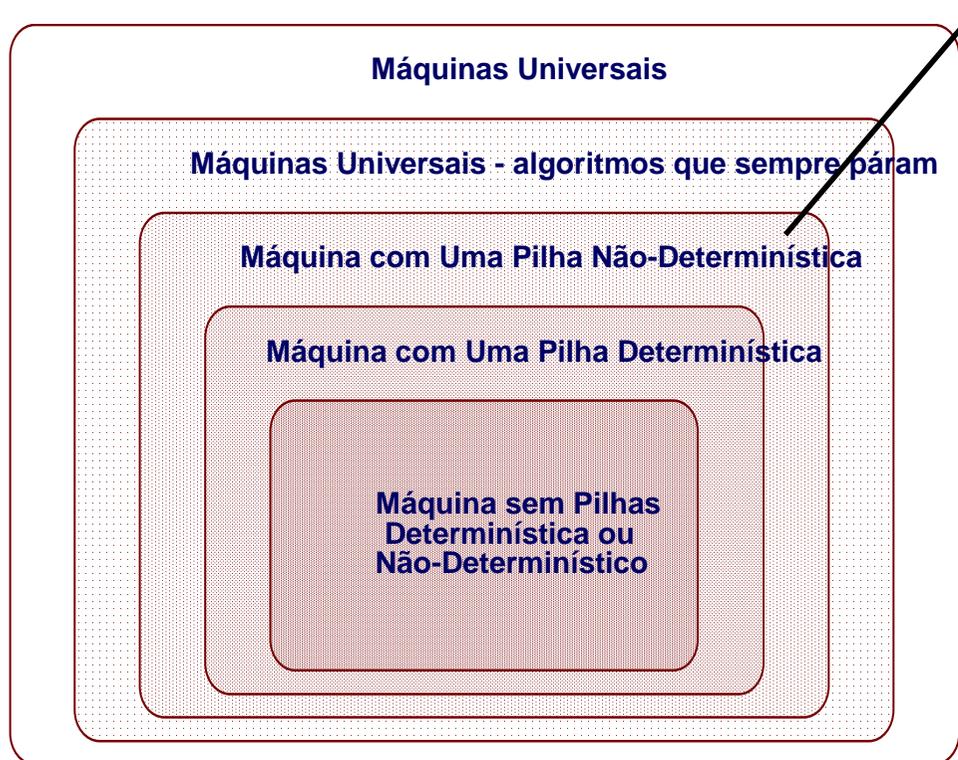
- tempo de reconhecimento de uma entrada é diretamente proporcional ao dobro do tamanho da entrada

### **Formalismos**

Gramática livre de contexto  
Autômato à pilha

# Hierarquia de Classes de Máquinas

## *Linguagens Livres do Contexto*

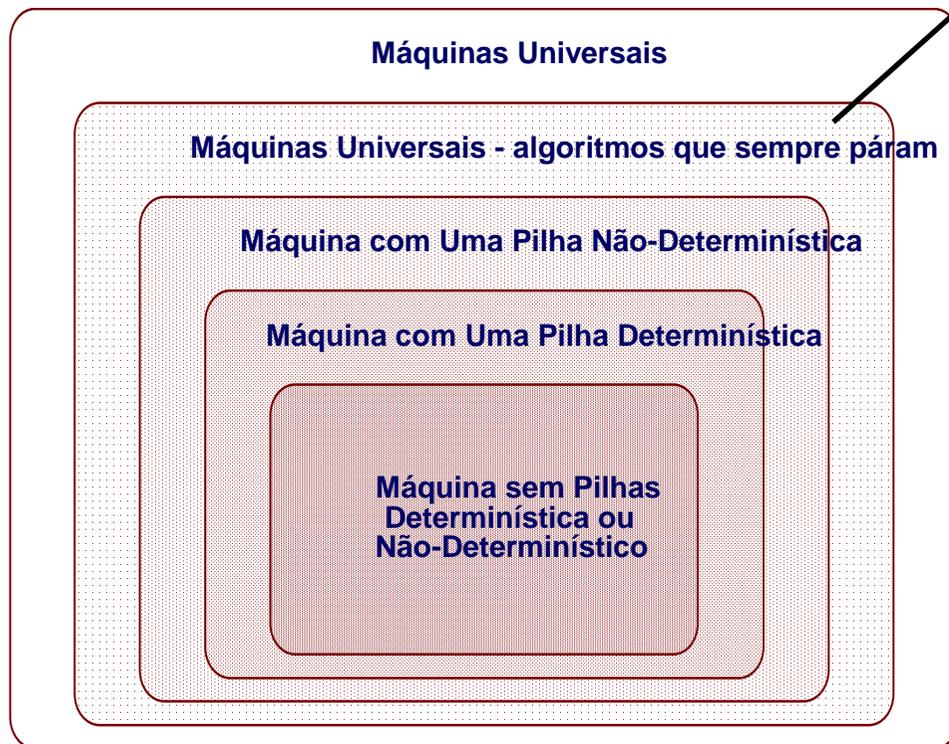


- Constituem uma classe de fundamental importância, pois incluem linguagens de programação como Algol e Pascal
- Algumas linguagens muito simples não pertencem a essa classe de linguagens como:  
$$\text{Tripto\_Bal} = \{ a^n b^n c^n \mid n > 0 \}$$
$$\text{Palavra\_Palavra} = \{ ww \mid w \text{ é palavra sobre os símbolos } a \text{ e } b \}$$
- Os melhores algoritmos de reconhecimento conhecidos possuem tempo de processamento proporcional ao tamanho da entrada elevado ao cubo

### **Formalismos**

Gramática livre de contexto  
Autômato à pilha

# Hierarquia de Classes de Máquinas

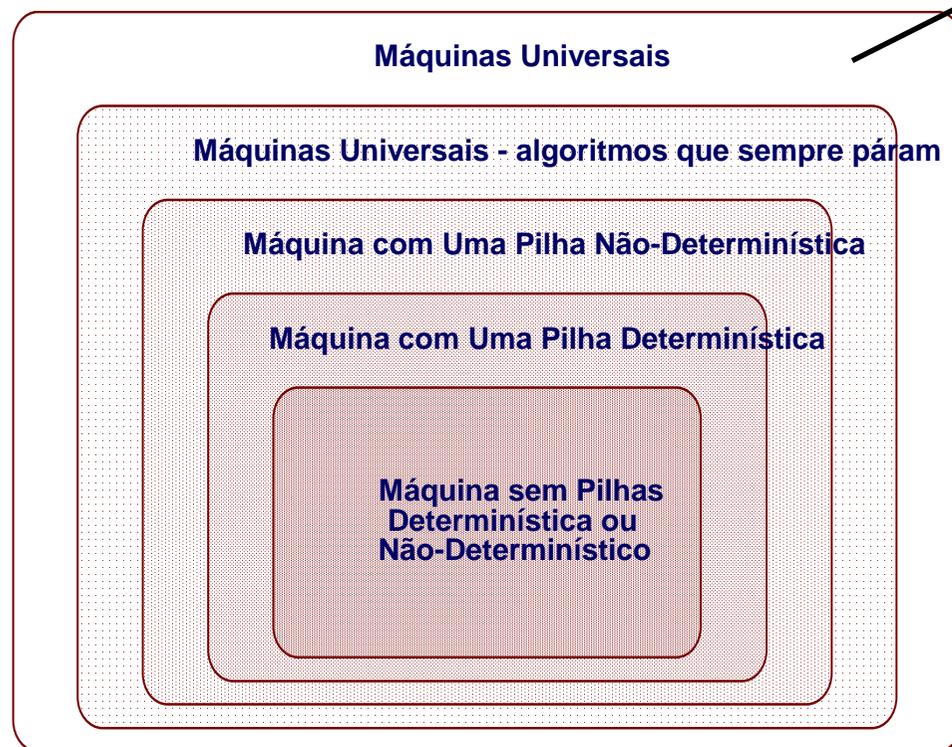


## *Linguagens Recursivas*

- 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

**Formalismos**  
MT e MP

# Hierarquia de Classes de Máquinas



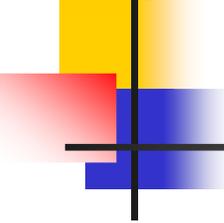
*Linguagens  
Recursivamente*

*Enumeráveis*

- Correspondem à classe de todas as linguagens que podem ser reconhecidas mecanicamente.

## **Formalismos**

MT, Máquina com duas ou mais pilhas  
Autômato com duas pilhas



# Outros Modelos de Máquinas Universais

---

- Exercícios Propostos
  - 3.2, 3.7, 3.24 e 3.25