# Universidade Federal de Alfenas

## Linguagens Formais e Autômatos

### Aula 15 – Máquinas de Turing (parte 2)

humberto@bcc.unifal-mg.edu.br

# Última Aula

- Uma Máquina de Turing (MT) possui:
  - uma fita infinita para representar a sua memória ilimitada;
  - Um cabeçote, para ler, e escrever na memória/fita;
  - Um controle;
  - Capacidade de movimentar-se para dois lados:
    - Direita
    - Esquerda

# Última aula

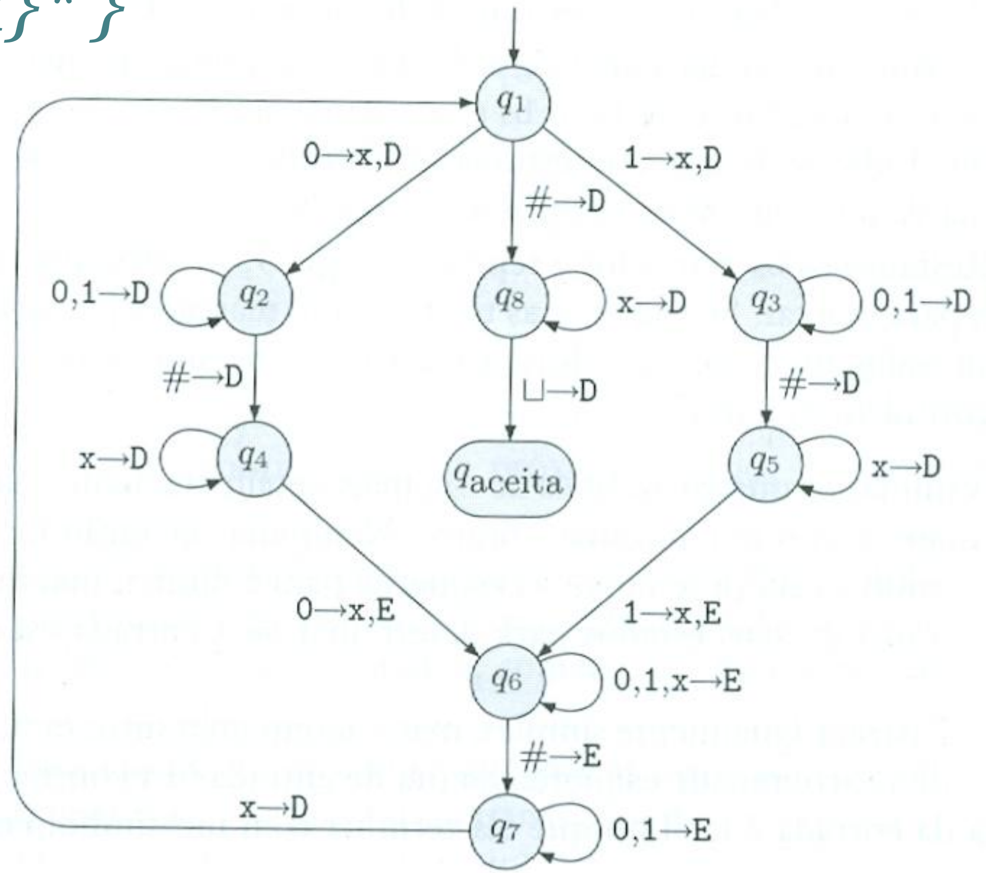
- Formalismo das MT

Uma ***máquina de Turing*** é uma 7-upla $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{aceita}}, q_{\text{rejeita}})$, onde $Q, \Sigma, \Gamma$ são todos conjuntos finitos e

1. $Q$ é o conjunto de estados.
2. $\Sigma$ é o alfabeto de entrada que não contém o símbolo especial ***branco*** $\sqcup$,
3. $\Gamma$ é o alfabeto da fita, onde $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$,
4. $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\mathrm{E}, \mathrm{D}\}$ é a função de transição,
5. $q_0$ é o estado inicial,
6. $q_{\text{aceita}} \in Q$ é o estado de aceitação,
7. $q_{\text{rejeita}} \in Q$ é o estado de rejeição.

# Última aula

- Exemplo de uma MT que reconhece uma linguagem que não é Livre do Contexto:
  - $L=\{ w\#w \mid w \in \{0,1\}^* \}$



# Variantes de Máquinas de Turing

# Variantes de Máquinas de Turing

- Existem várias **versões que estendem a Máquinas de Turing**, por exemplo:

  - MT com **Múltiplas fitas**;
  - MT que permitem o **Não-Determinismo**;

# Variantes de Máquinas de Turing

- Existem várias versões que estendem a Máquinas de Turing, por exemplo:
  - MT com Múltiplas fitas;
  - MT que permitem o Não-Determinismo;
- O modelo original, e suas variações razoáveis possuem o mesmo poder computacional;

# Variantes de Máquinas de Turing

- Existem várias versões que estendem a Máquinas de Turing, por exemplo:
  - MT com Múltiplas fitas;
  - MT que permitem o Não-Determinismo;
- O modelo original, e suas variações razoáveis possuem o mesmo poder computacional;
- Na teoria da computação, isso é chamado de Robustez.

# Variação #1

MT com cabeçote que pode ficar parado

Função de transição da MT padrão: $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$

Função de transição da MT estendida: $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D, P\}$

# MT com cabeçote que pode ficar parado

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D, P\}$$

- Essa característica poderia permitir as MT reconhecer linguagens adicionais?... incrementando o poder do modelo computacional

# MT com cabeçote que pode ficar parado

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D, P\}$$

- Essa característica poderia permitir as MT reconhecer linguagens adicionais?... incrementando o poder do modelo computacional

- Não!
  - Podemos converter QUALQUER MT com a característica adicional, em uma MT padrão;

# MT com cabeçote que pode ficar parado

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D, P\}$$

- Essa característica poderia permitir as MT reconhecer linguagens adicionais?... incrementando o poder do modelo computacional

- **Não!**
  - Podemos converter QUALQUER MT com a característica adicional, em uma MT padrão;
  - Fazemos isso **substituindo cada transição com "P", por duas transições**, uma que move para a direita, e uma segunda que move para a esquerda.

# Variação #2

MT com Múltiplas Fitas

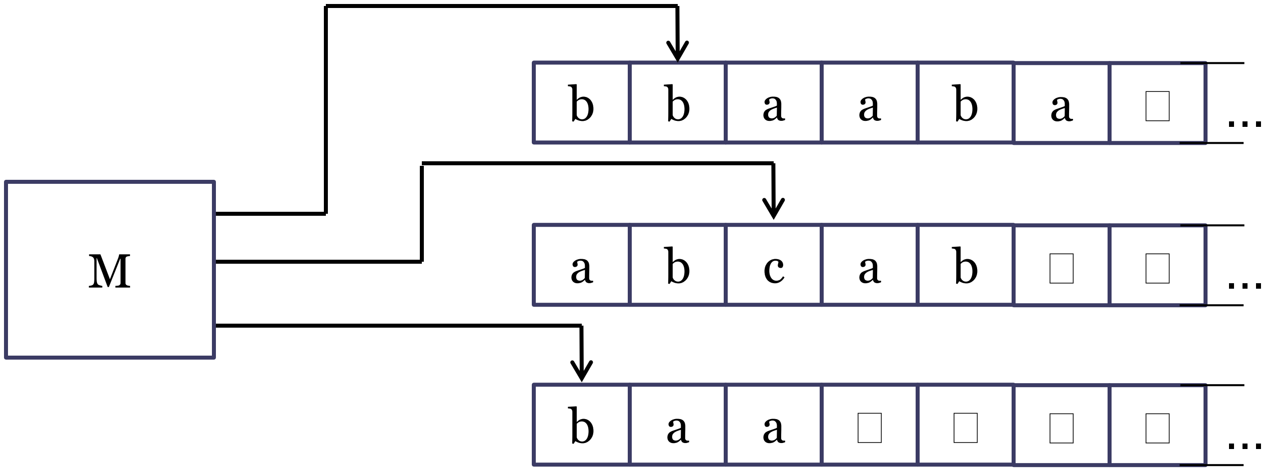
Função de transição da MT padrão: $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$

Função de transição da MT estendida: $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{E, D\}^k$

# MT com Múltiplas Fitas

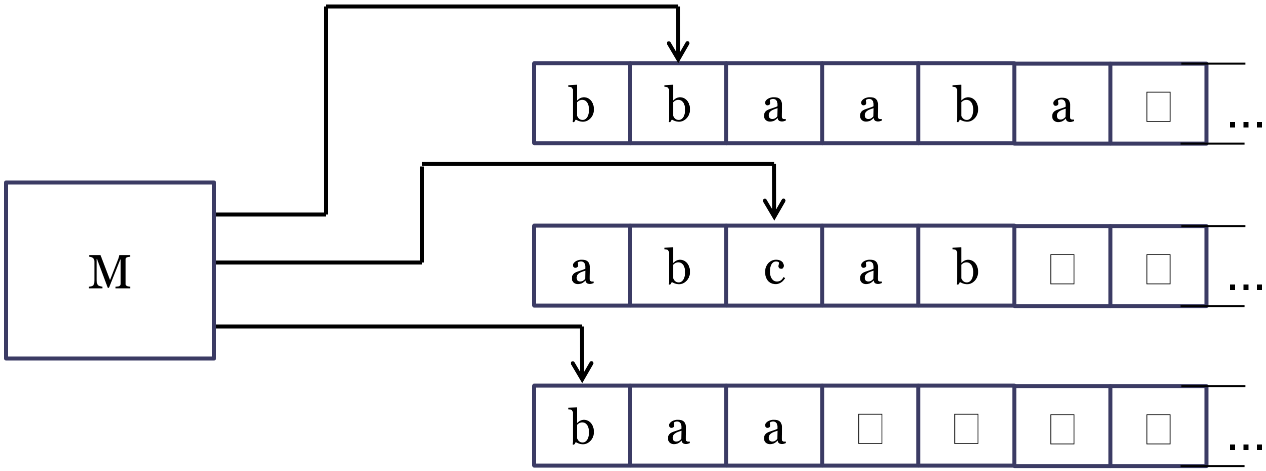
$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{E, D\}^k$$

- A função de transição é modificada para permitir ler, escrever e mover as cabeças em todas as fitas, simultaneamente.



# MT com Múltiplas Fitas

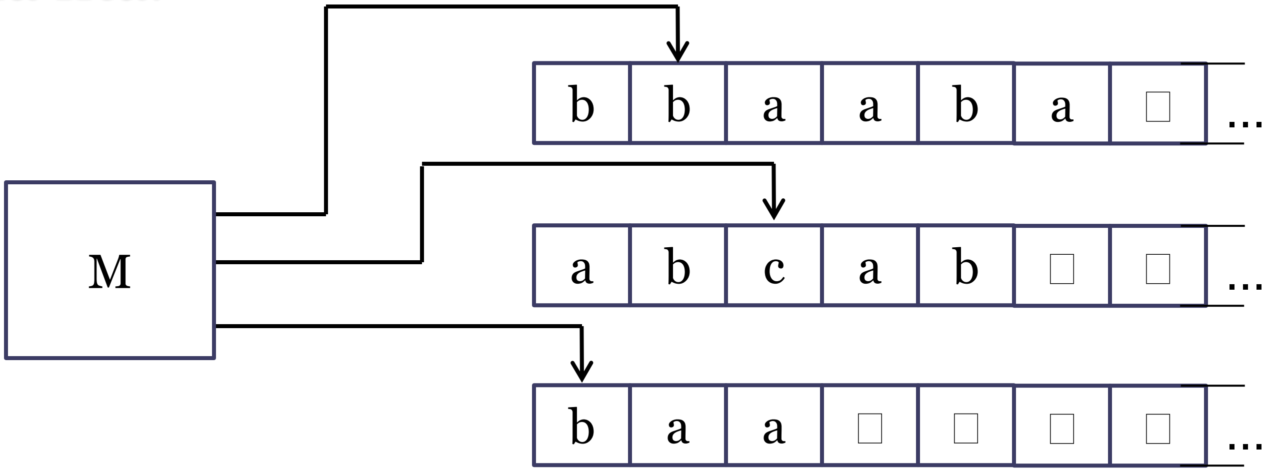
- A princípio, elas parecerem ser mais poderosas computacionalmente, mas podemos mostrar que elas reconhecem o mesmo conjunto de linguagens.



# MT com Múltiplas Fitas

- A princípio, elas parecerem ser mais poderosas computacionalmente, mas podemos mostrar que elas reconhecem o mesmo conjunto de linguagens.

- **Para isso, mostramos como converter uma MT com Múltiplias Fitas M em uma equivalente S, com apenas uma fita.**



# MT com Múltiplas Fitas

- *S* simula o efeito das *k* fitas armazenando sua informação e uma única fita.

# MT com Múltiplas Fitas

- S simula o efeito das k fitas armazenando sua informação e uma única fita.

- A fita de *S* possui um novo símbolo # para separar o conteúdo das fitas de M.

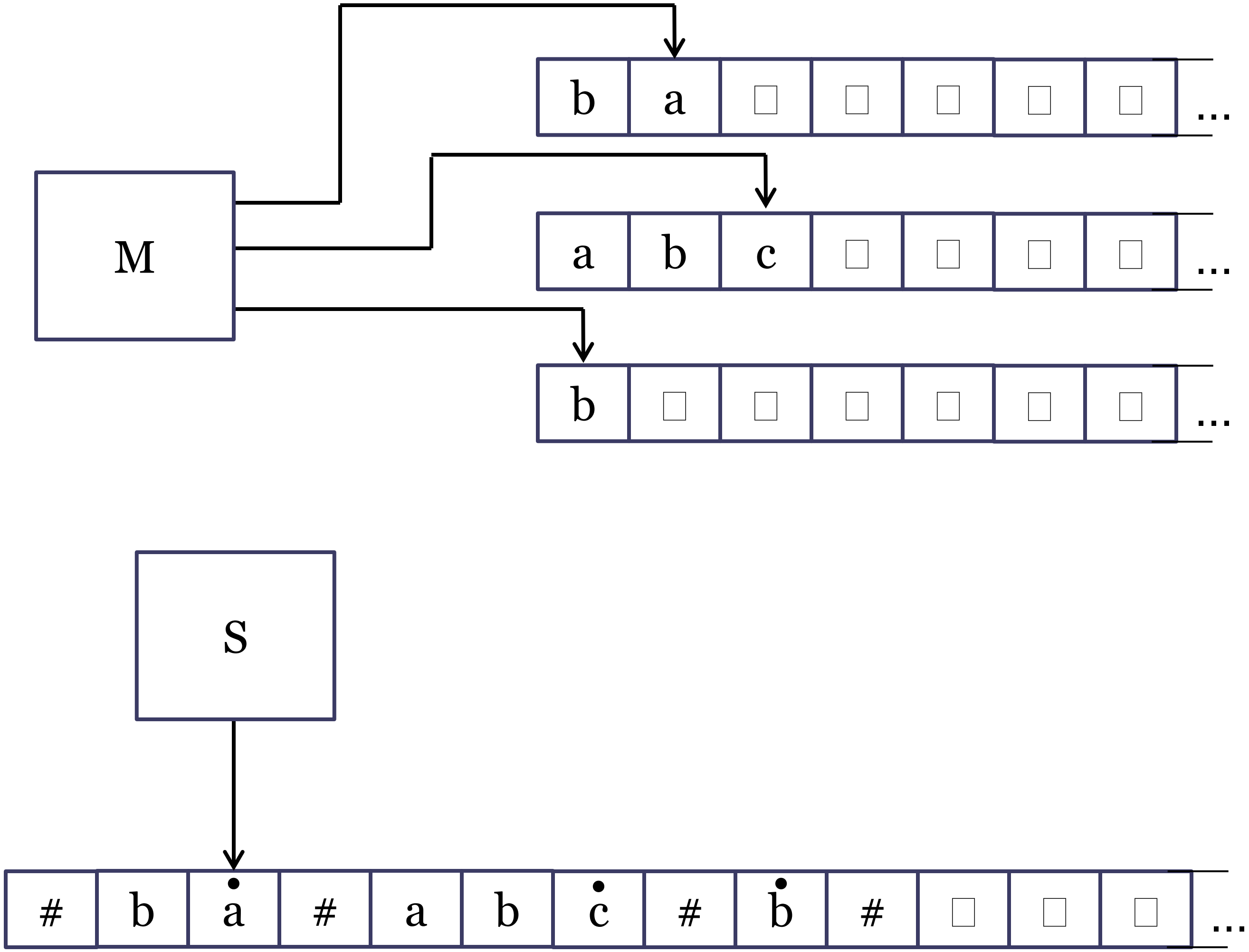
# MT com Múltiplas Fitas

- S simula o efeito das k fitas armazenando sua informação e uma única fita.

- A fita de S possui um novo símbolo # para separar o conteúdo das fitas de M.

- ***S*** **também precisa manter o controle dos "contadores de programa"**. Para isso, ela usa símbolos especiais, com um ponto acima do símbolo original.

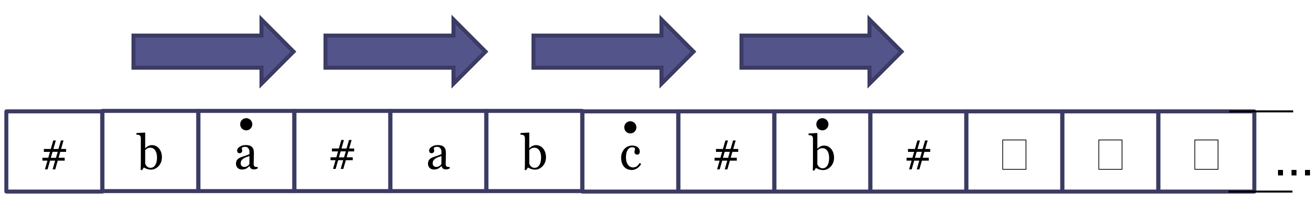
# MT com Múltiplas Fitas



# MT com Múltiplas Fitas

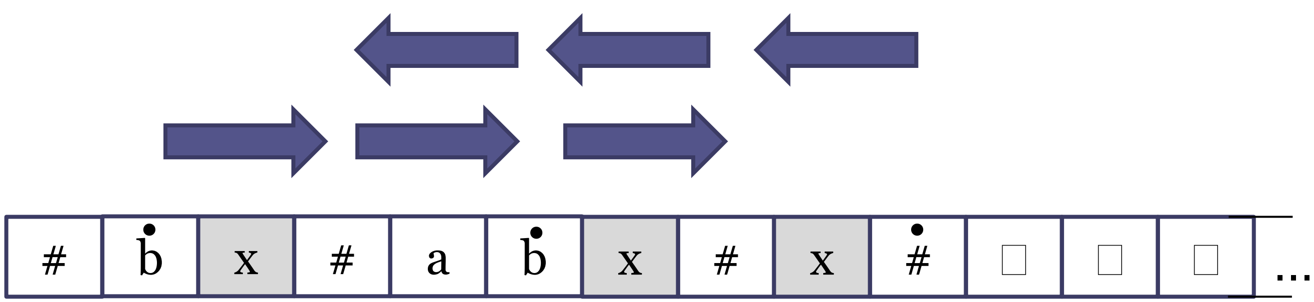
- Explicação geral sobre o funcionamento de *S*:

  - Para simular um único movimento de *M*, *S* faz uma varredura na sua fita, desde o primeiro #, até o (k+1)-ésimo #, de modo a determinar os símbolos sobre as cabeças virtuais;



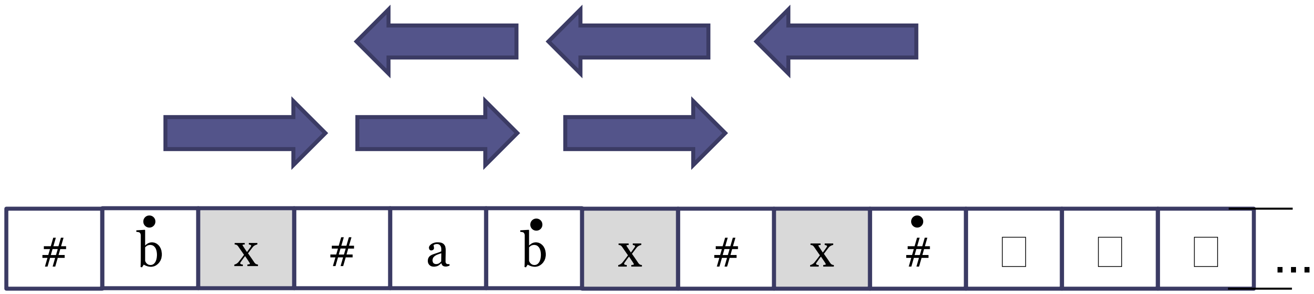
# MT com Múltiplas Fitas

- <u>Explicação geral sobre o funcionamento de *S*:</u>
  - Sabendo os símbolos sobre todos os "contadores de programa", *S* faz uma nova passagem atualizando as fitas simuladas em *S*, como *M* faria com múltiplas fitas;



# MT com Múltiplas Fitas

- Explicação geral sobre o funcionamento de *S*:
  - *S* precisa de mais memória relacionada com os estados, se comparada com M.



# Variação #2

MT Não-Determinística

Função de transição da MT padrão: $\delta : Q \times \Gamma \to Q \times \Gamma \times \{E, D\}$

Função de transição da MT estendida: $\delta : Q \times \Gamma \to P(Q \times \Gamma \times \{E, D\})$

# MT não-determinística

- A computação de uma máquina de Turing não-determinística (MTND) é uma árvore cujos ramos correspondem a diferentes possibilidades para a máquina;

# MT não-determinística

- A computação de uma máquina de Turing não-determinística (MTND) é uma árvore cujos ramos correspondem a diferentes possibilidades para a máquina;

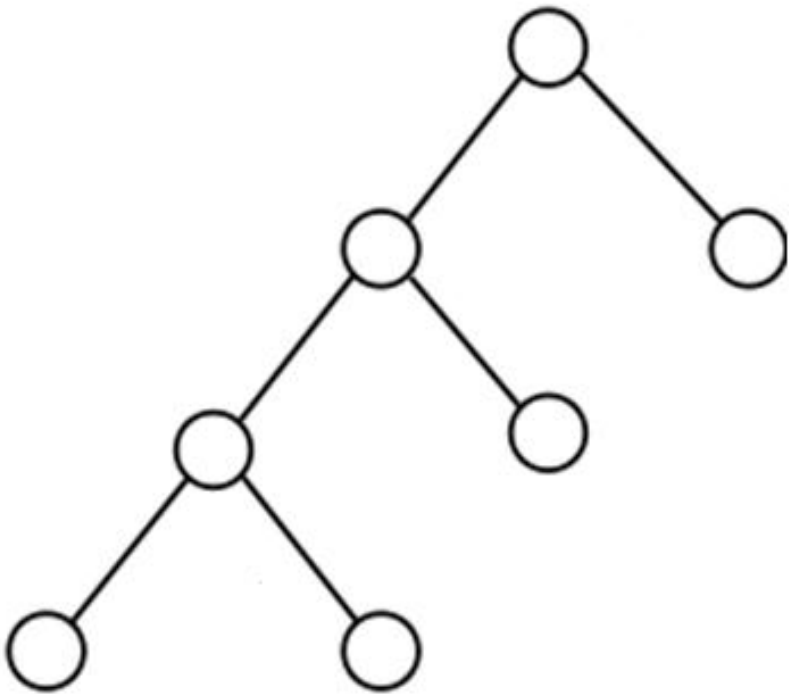
- Se algum ramo da computação leva ao estado de aceitação, a máquina aceita sua entrada;

# MT não-determinística

- A computação de uma máquina de Turing não-determinística (MTND) é uma árvore cujos ramos correspondem a diferentes possibilidades para a máquina;
- Se algum ramo da computação leva ao estado de aceitação, a máquina aceita sua entrada;
- Podemos simular uma MTND $M$, através de uma MT $S$, determinística.

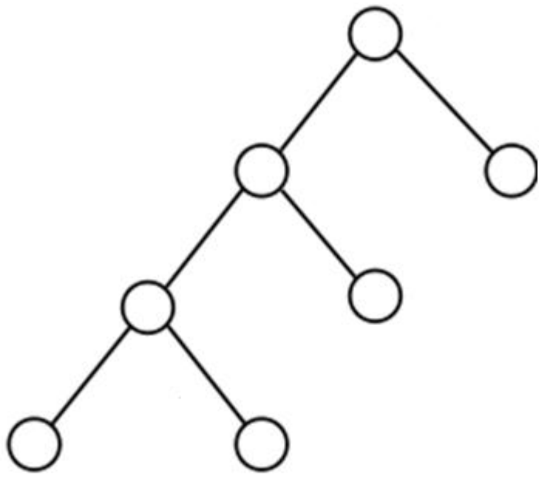
# MT não-determinística

- Vemos a computação de M sobre uma entrada $w$ como uma <u>árvore de possibilidades</u>:

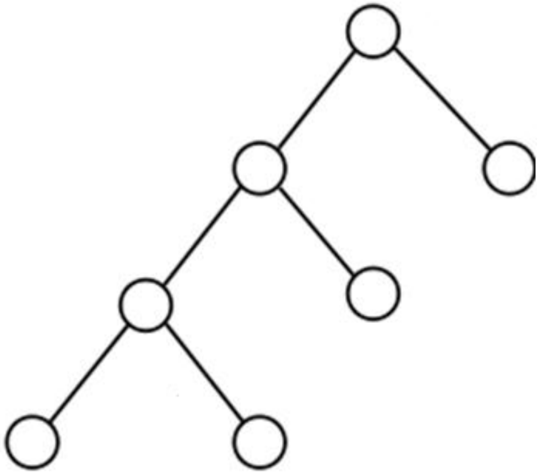


# MT não-determinística

- Cada ramo da árvore é uma configuração de *M* (controle + fita);
- A MT *S* busca um estado de aceitação nesta árvore de possibilidades;

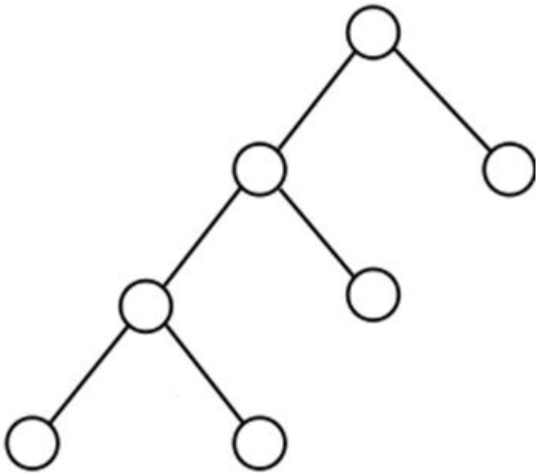


# MT não-determinística



- Uma idéia tentadora é fazer a máquina *S* tentar uma busca em profundidade na árvore de possibilidades;

# MT não-determinística



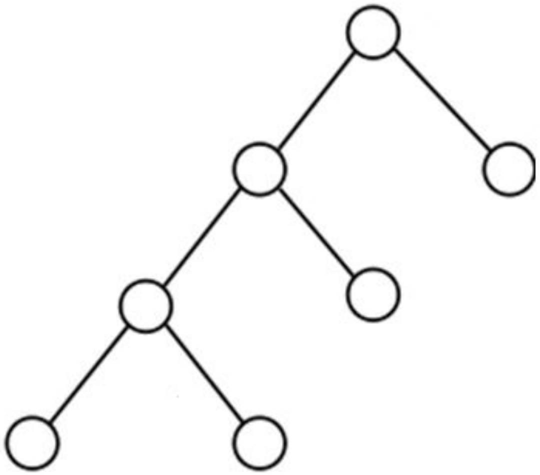
- Uma idéia tentadora é fazer a máquina S tentar uma busca em profundidade na árvore de possibilidades;

- O **problema** de tal estratégia é a possibilidade de **uma das ramificações não ter fim**...

    - A máquina (em profundidade), não encontraria resposta, quando na verdade existe a possibilidade de outro ramo ser finito, e levando a um estado de aceitação.

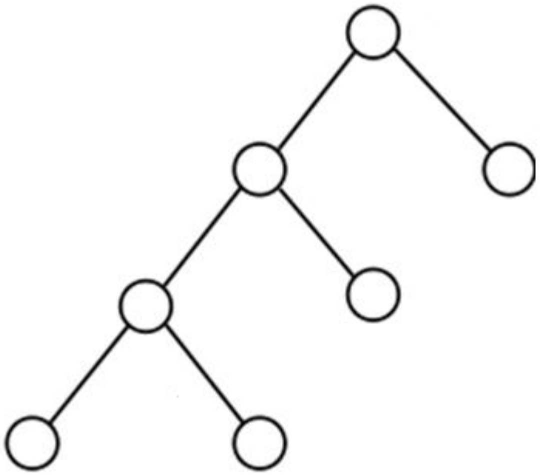
# MT não-determinística

- A solução então é **computar no modelo de busca em largura**!
    - Todos os nós do nível *k*, são explorados antes de qualquer nó do nível *k+1*;



# MT não-determinística

- A solução então é computar no modelo de busca em largura!
  - Todos os nós do nível *k*, são explorados antes de qualquer nó do nível *k+1*;
- Este método garante que *S* visitará todo nó na árvore até que ela encontre uma configuração de aceitação, se ela existir.



# MT não-determinística

- A MT simuladora *S* possui 3 fitas:
  - A fita 1 possui a cadeia de entrada (nunca é alterada);

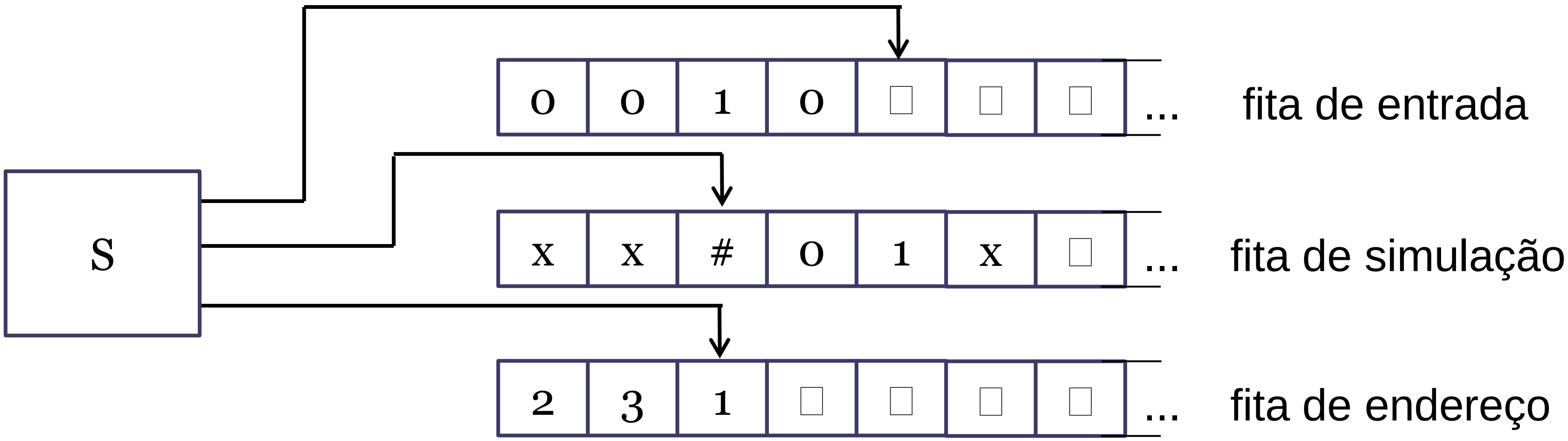
# MT não-determinística

- A MT simuladora *S* possui 3 fitas:
  - A fita 1 possui a cadeia de entrada (nunca é alterada);
  - A fita 2 mantém uma cópia da fita de *M* em algum ponto de sua computação;

# MT não-determinística

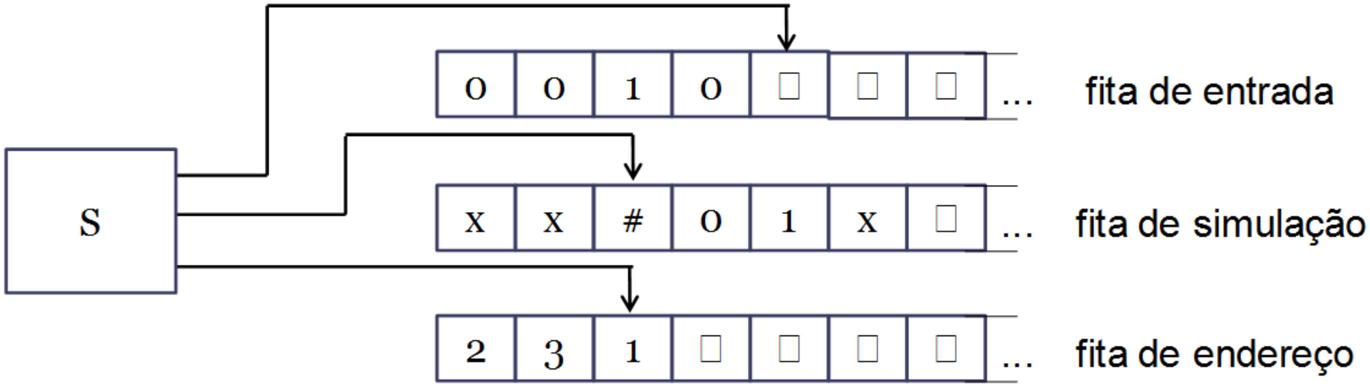
- A MT simuladora *S* possui 3 fitas:
  - A fita 1 possui a cadeia de entrada (nunca é alterada);
  - A fita 2 mantém uma cópia da fita de *M* em algum ponto de sua computação;
  - A fita 3 mantém o registro da posição de *S* na árvore de *M*.

# MT não-determinística



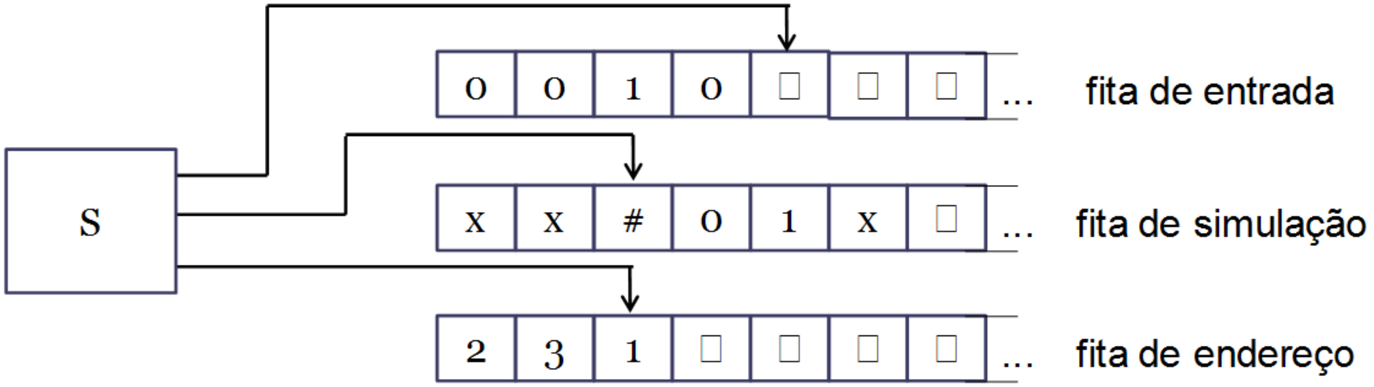
# Como a simulação da MTND funciona

1. Inicialmente, a fita 1 contém a entrada $w$ e as fitas 2 e 3 estão vazias;



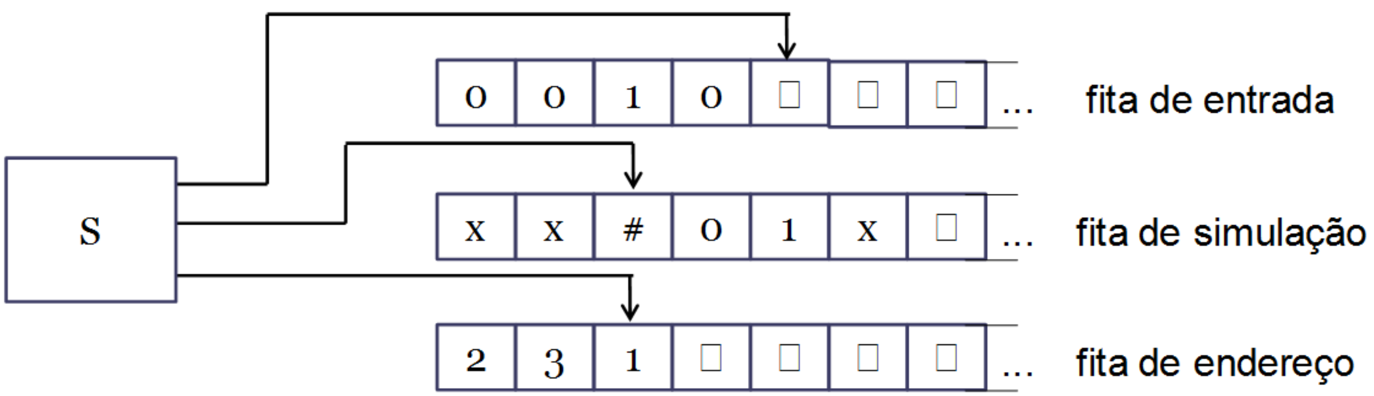
# Como a simulação da MTND funciona

1. Inicialmente, a fita 1 contém a entrada $w$ e as fitas 2 e 3 estão vazias;
2. Copie a fita 1 para a fita 2;



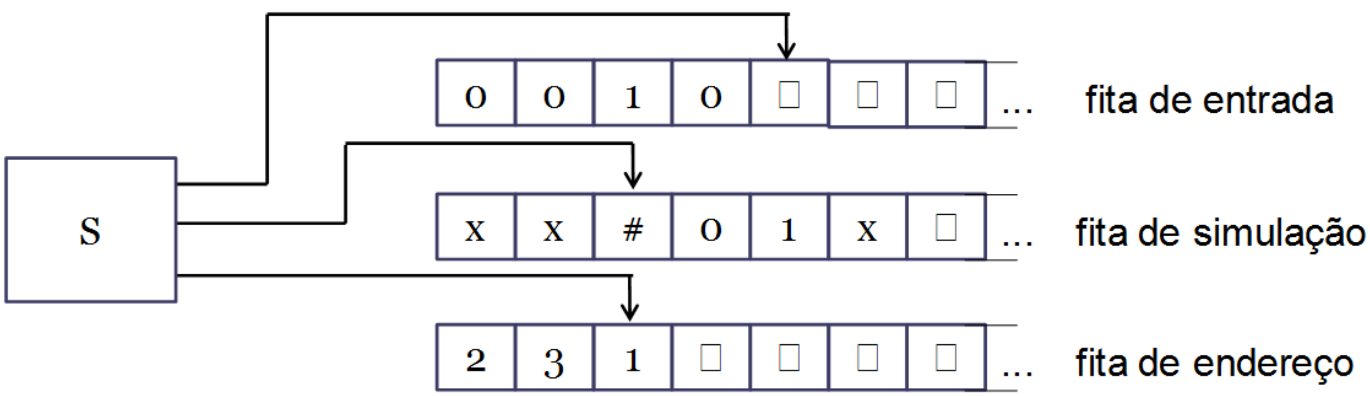
# Como a simulação da MTND funciona

1. Inicialmente, a fita 1 contém a entrada $w$ e as fitas 2 e 3 estão vazias;
2. Copie a fita 1 para a fita 2;
3. Use a fita 2 para simulação, seguindo a ordem de escolha das transições na fita 3.



# Como a simulação da MTND funciona

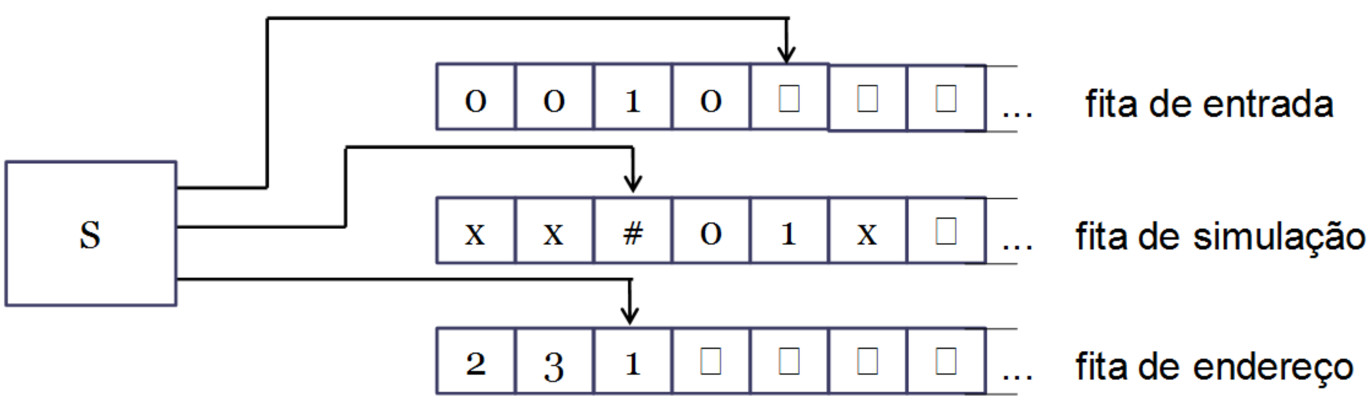
1. Inicialmente, a fita 1 contém a entrada $w$ e as fitas 2 e 3 estão vazias;
2. Copie a fita 1 para a fita 2;
3. Use a fita 2 para simulação, seguindo a ordem de escolha das transições na fita 3.
4. Se $w$ não for reconhecida, atualize a fita 3 (para um valor válido), e volte ao passo 2;



# Como a simulação da MTND funciona

Se não existir mais possibilidade de atualização neste nível, a máquina $S$ precisa consumir mais um símbolo branco na fita 3 e avançar mais um nível na busca em largura.

1. Inic… …da $w$ e as fitas 2 e 3 es…
2. Cop…
3. Use … …lo a ordem de escolha das transições na … 3.
4. Se $w$ não for reconhecida, atualize a fita 3 (para um valor válido), e volte ao passo 2;



# Próxima Aula

- A Tese de Church-Turing;

- O Problema da Parada;

  - O Problema da Correspondência de *Post* (PCP)

# Bibliografia


- SIPSER, Michael. Introdução à Teoria da Computação. 2a ed.:São Paulo, Thomson, 2007.




- VIEIRA, Newton José. Introdução aos Fundamentos da Computação: Linguagens e Máquinas. 1a ed.: Rio de Janeiro: Thomson, 2006.