ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 14

Cap 3.2 – Variantes de MT (Enumeradores)

Profa. Ariane Machado Lima
ariane.machado@usp.br

Aulas anteriores...

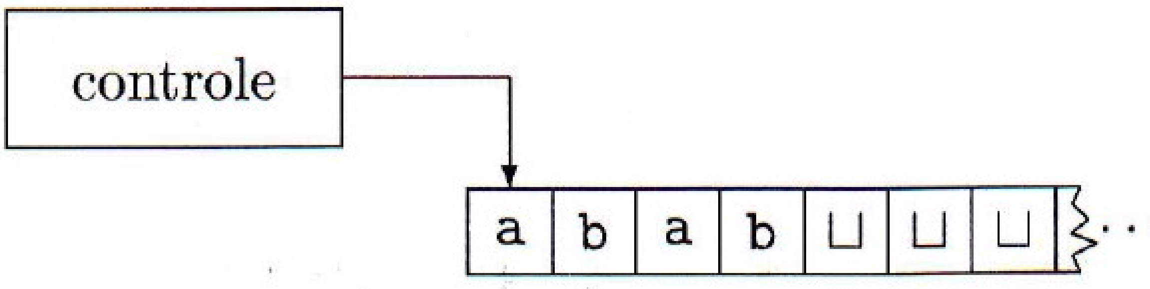
Máquinas de Turing – Definição formal

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

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

$\delta: Q' \times \Gamma \longrightarrow Q \times \Gamma \times \{E, D\}$, onde Q' é Q sem q_{aceita} e q_{rejeita}

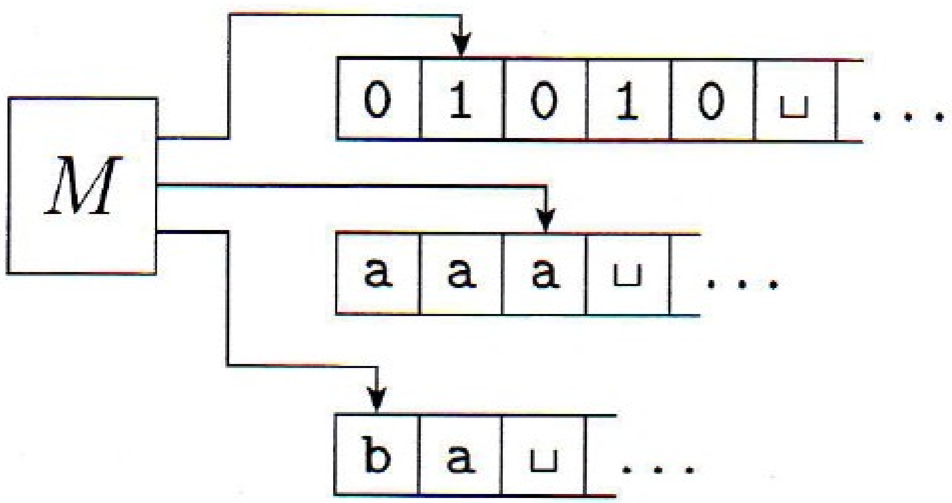
Máquinas de Turing



1. Uma máquina de Turing pode tanto escrever sobre a fita quanto ler a partir dela.
2. A cabeça de leitura-escrita pode mover-se tanto para a esquerda quanto para a direita.
3. A fita é infinita.
4. Os estados especiais para rejeitar e aceitar fazem efeito imediatamente.

3.2 – Variantes de Máquinas de Turing
(Dispositivos EQUIVALENTES)

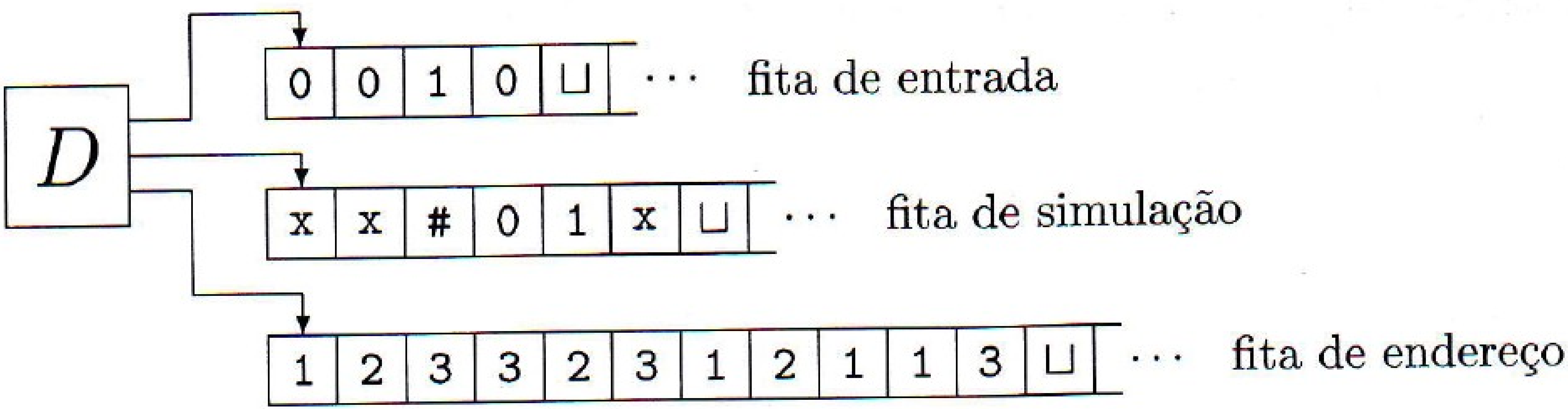
Máquinas de Turing Multifita



Máquinas de Turing Não-Determinísticas

$$\delta : Q \times \Gamma \longrightarrow \mathcal{P}(Q \times \Gamma \times \{E, D\}).$$

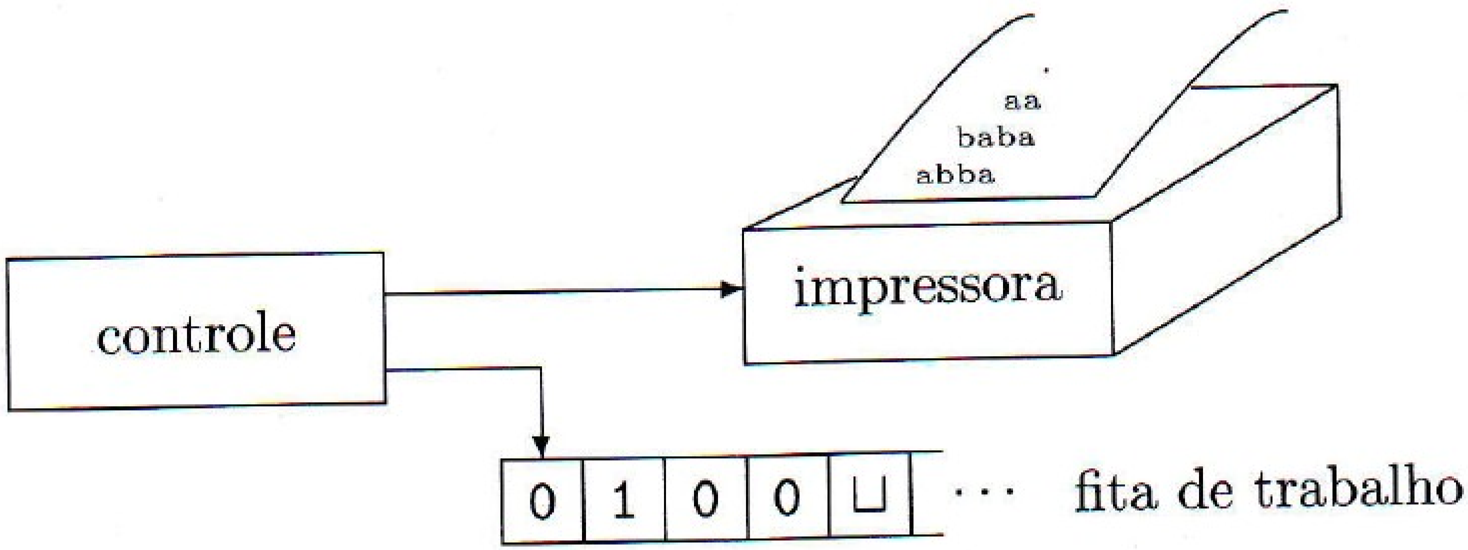
Simulação de uma MTND por uma MT Determinística:



Uma cadeia é aceita se ALGUM ramo da computação a aceita
Uma cadeia não é aceita se NENHUM ramo da computação a aceita (NÃO basta um ramo rejeitar)

Aula de hoje

Enumeradores



Duas fitas, sendo uma só de escrita (impressora)

Começa com a fita impressora em branco

Imprime cadeias da linguagem, em qualquer ordem, possivelmente com repetição

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (<=)

Temos um enumerador E

M funciona assim: dada uma cadeia w

-
-
-

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (<=)

Temos um enumerador E

M funciona assim: dada uma cadeia w

- roda E e compara a cadeia enumerada com w
- se for igual, aceita w
- rejeita w se acabar a enumeração

M aceita as cadeias enumeradas por E

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (<=)

Temos um enumerador E

Uma MT M que utiliza E funciona assim: dada uma cadeia w

- roda E e compara a cadeia enumerada com w
- se for igual, aceita w
- rejeita w se acabar a enumeração

Se w pertence a L, será aceita

Se w não pertence a L,

TEOREMA 3.21 ....

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (<=)

Temos um enumerador E

Uma MT M que utiliza E funciona assim: dada uma cadeia w

- roda E e compara a cadeia enumerada com w
- se for igual, aceita w
- rejeita w se acabar a enumeração

Se w pertence a L, será aceita

Se w não pertence a L,

M rejeita se L for finita

M não pára se L for infinita

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

-
-
  -
  -

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja s_1, s_2, s_3, … sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

-
-

TEOREMA 3.21 ....

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja s_1, s_2, s_3, … sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

- Ignore a entrada
- para i = 1, 2, 3, …
  - Rode M sobre s_i
  - Imprima s_i se for aceita

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja s_1, s_2, s_3, … sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

- Ignore a entrada
- para i = 1, 2, 3, …
  - Rode M sobre s_i
  - Imprima s_i se for aceita

Problema?

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja $s_1, s_2, s_3, \ldots$ sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

- Ignore a entrada
- para i = 1, 2, 3, …

    Rode M sobre s_i

    Imprima s_i se for aceita

Problema?

E se M entrar em loop?

TEOREMA 3.21 ....

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja s_1, s_2, s_3, … sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

- Ignore a entrada
- para i = 1, 2, 3, …
  - Rode M sobre s_i
  - Imprima s_i se for aceita

Problema?

E se M entrar em loop? Travará o enumerador...

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja s_1, s_2, s_3, … sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

- ~~Ignore a entrada~~
- ~~para i = 1, 2, 3, …~~
  - ~~Rode M sobre~~ $\cancel{s_i}$
  - ~~Imprima~~ $\cancel{s_i}$ ~~se for aceita~~

Problema?

E se M entrar em loop? Travará o enumerador...

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja s_1, s_2, s_3, … sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

- Ignore a entrada
- para i = 1, 2, 3, …
  - Rode i passos de M sobre s_1, ..., s_i
  - Imprima as s_j que foram aceitas

TEOREMA 3.21 ....

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Prova: (=>) Temos uma Máquina de Turing M

O enumerador E funciona assim:

Seja $s_1, s_2, s_3, \ldots$ sequências de Σ^* (ordem crescente de comprimento e ordem lexicográfica)

- Ignore a entrada
- para i = 1, 2, 3, …

  Rode i passos de M sobre $s_1, \ldots, s_i$

  Imprima as s_j que foram aceitas

Se w pertence a L, w é impressa um número infinito de vezes

TEOREMA 3.21

Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Por isso linguagens Turing-reconhecíveis são também chamadas linguagens recursivamente enumeráveis