# ITC: Introdução à Teoria da Computação

Marcos Castilho

DInf/UFPR

12 de julho de 2021

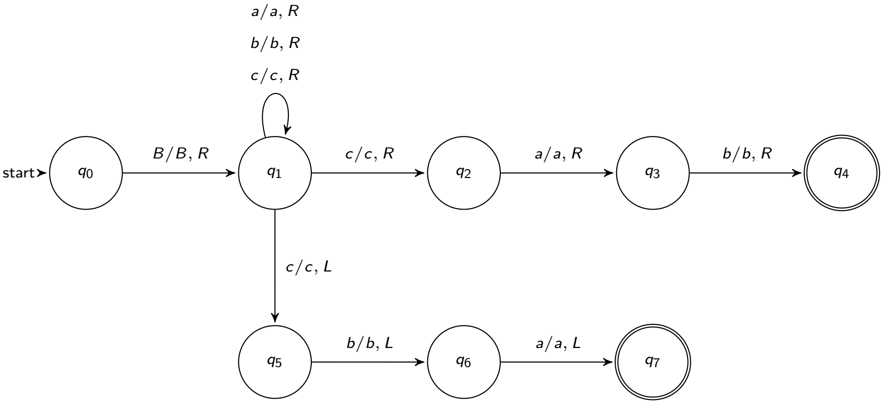
# Máquinas de Turing Não Determinísticas

Definição: Uma Máquina de Turing Não Determinística (MTND) é uma sextupla $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, onde:

- $Q, \Sigma, \Gamma, \delta, q_0, F$ são como na MT padrão;
- $\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$.

Uma entrada é aceita por uma MTND se existe uma computação que termina em um estado final.

# Exemplo



$a/a, R$
$b/b, R$
$c/c, R$
start
$q_0$
$B/B, R$
$q_1$
$c/c, R$
$q_2$
$a/a, R$
$q_3$
$b/b, R$
$q_4$
$c/c, L$
$q_5$
$b/b, L$
$q_6$
$a/a, L$
$q_7$

# Equivalência para MT

Podemos ordenar as computações assim:

- Seja $N$ o maior número de transições para um par de estado e símbolo na fita;
- $\delta(q_i, x)$ define assim $N$ transições possíveis para cada par $(q_i, x)$;
- Pode haver repetições
- $(m_1, m_2, \ldots, m_k)$ de valores de 1 a $N$ define uma computação que pode ter $k$ ou menos transições;
- Suponha que a transição $\delta(q_i, x)$ foi numerada $m_j$. Se a $(j-1)$-ésima transição deixou a máquina no estado $q_i$ e símbolo $x$ na fita, então $\delta(q_i, x)$ pode ser executada, e $m_j$ será a $j$-ésima posição em $(m_1, \ldots, m_j, \ldots, m_k)$.

Para a MTND do exemplo anterior, temos o seguinte:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $q_0$ | $B$ | 1 | $q_1, B, R$ | $q_2$ | $a$ | 1 | $q_3, a, R$ |
| | | 2 | $q_1, B, R$ | | | 2 | $q_3, a, R$ |
| | | 3 | $q_1, B, R$ | | | 3 | $q_3, a, R$ |
| $q_1$ | $a$ | 1 | $q_1, a, R$ | $q_3$ | $b$ | 1 | $q_4, b, R$ |
| | | 2 | $q_1, a, R$ | | | 2 | $q_4, b, R$ |
| | | 3 | $q_1, a, R$ | | | 3 | $q_4, b, R$ |
| $q_1$ | $b$ | 1 | $q_1, b, R$ | $q_5$ | $b$ | 1 | $q_6, b, L$ |
| | | 2 | $q_1, b, R$ | | | 2 | $q_6, b, L$ |
| | | 3 | $q_1, b, R$ | | | 3 | $q_6, b, L$ |
| $q_1$ | $c$ | 1 | $q_1, c, R$ | $q_6$ | $a$ | 1 | $q_7, a, L$ |
| | | 2 | $q_2, c, R$ | | | 2 | $q_7, a, L$ |
| | | 3 | $q_5, c, L$ | | | 3 | $q_7, a, L$ |

Computações para a palavra *acab* e as sequências:

- (1,1,1,1,1):
  $q_0 BacabB$ (1) $\vdash Bq_1 acabB$ (1) $\vdash Baq_1 cabB$ (1) $\vdash Bacq_1 abB$ (1) $\vdash Bacaq_1 bB$ (1) $\vdash Bacabq_1 B$ (1)
- (1,1,2,1,1):
  $q_0 BacabB$ (1) $\vdash Bq_1 acabB$ (1) $\vdash Baq_1 cabB$ (2) $\vdash Bacq_2 abB$ (1) $\vdash Bacaq_3 bB$ (1) $\vdash Bacabq_4 B$ (1)
- (2,2,3,3,1):
  $q_0 BacabB$ (2) $\vdash Bq_1 acabB$ (2) $\vdash Baq_1 cabB$ (3) $\vdash Bq_5 acabB$

# Equivalência com MT

Assim, uma MTD pode simular uma MTND!

- Seja $M$ uma MTND, $M = (Q, \Sigma, \Gamma, \delta, q_0)$ que aceita por parada.
- Pode-se construir uma MTD com três fitas que também aceita por parada.
- Basta simular a execução possível de toda sequência $(m_1, \ldots, m_k)$ da MTND $M$.

# Equivalência com MT

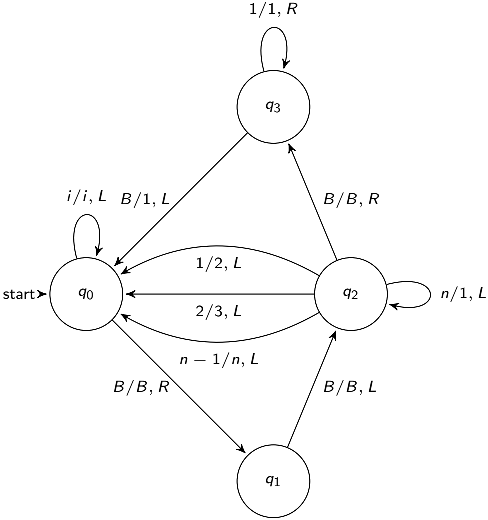
$M'$ é assim projetada:

1. Escrever uma sequência de inteiros entre 1 e $N$ na fita 3;
2. Copiar a entrada da fita 1 na fita 2;
3. Simular $M$ na fita 2 de acordo com a sequência da fita 3;
4. Se a simulação para, a computação de $M'$ para e a entrada é aceita;
5. Uma nova sequência é gerada na fita 3 e os passos 2-5 são repetidos.

# Problema

Como garantir que toda sequência $(m_1, \ldots, m_k)$ seja gerada?

# Solução



## Construção de $M'$

Sejam:

- $\Sigma_{M'} = \Sigma$
- $\Gamma_{M'} = \{(x, \#x)\} \cup \{1, 2, \ldots, N\}$
- $\delta$ será definido assim:

# Definição de $\delta$

- Obtenha as transições que constituem a sequência na fita 3, a partir da MT apresentada;
- Mantenha os outros cabeçotes parados;
- Faça uma cópia da entrada que está na fita 1 na fita 2, mantendo a fita 3 inalterada, com o cabeçote parado;
- Traduza a computação da máquina original em uma simulação na fita 2;
- A computação de $M'$ para se a de $M$ para para alguma sequência;
- Antes de continuar, a computação simulada na fita 2 deve ser apagada e os cabeçotes 2 e 3 são retornados para a esquerda.

# Curiosidade

- A máquina original de Alan Turing (1936): era determinística, a fita era infinita para os dois lados e tinha um único cabeçote leitor.
- Turing foi condenado pelo governo britânico por homossexualidade em 1952.
- Em 2013, recebeu o perdão da rainha Elizabeth.

# Máquinas de Turing para enumerar linguagens

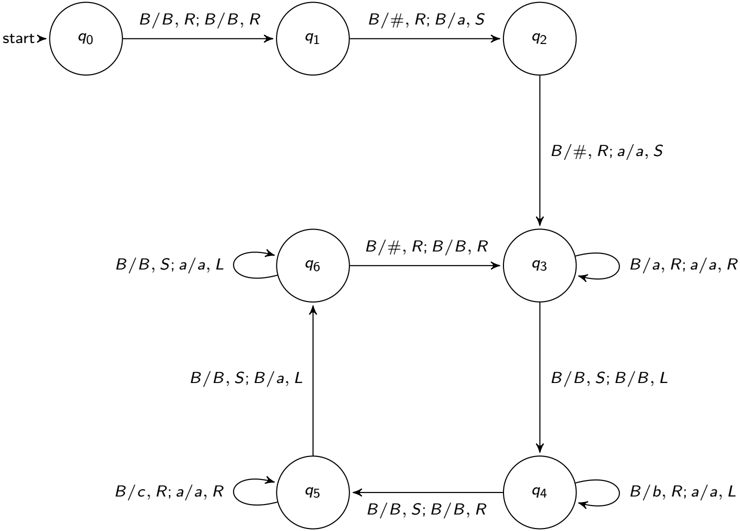
- Estas MT's produzem uma lista exaustiva com todos os elementos da linguagem;
- Estas MT's não têm entrada, apenas saída;
- Elas fazem a computação até que *todos* os elementos estejam na fita;
- Pode-se defini-las de várias maneiras, nós usaremos uma MT com $k$ fitas, $k \geq 2$;
- A fita 1 contém a saída, as outras são fitas de trabalho. Usa-se o símbolo $\#$ para separar os elementos na fita de saída.

# Definição

Uma MT com $k$ fitas $M = (Q, \Sigma, \Gamma, \delta, q_0)$ enumera uma linguagem $L$ se:

- A computação inicia com todas as fitas contendo brancos;
- Em cada transição o cabeçote da fita 1 (a fita de saída) permanece parada ou se move para a direita;
- Em qualquer ponto da computação a parte não branca da fita 1 tem uma das duas formas seguitnes:
  - $B\#u_1\#u_2\#\ldots\#u_k\#$, ou
  - $B\#u_1\#u_2\#\ldots\#u_k\#v$, onde $u_i \in L$ e $v \in \Sigma^*$.
- $u$ será escrito na fita 1 seguido e precedido por um $\#$ se, e somente se, $u \in L$.

# Exemplo: $L = \{a^i b^i c^i \mid i \geq 0\}$



# Teorema

Seja $L$ uma linguagem enumerada por uma máquina de Turing $M$. Então existe uma máquina de Turing $M'$ que enumera $L$ e na qual cada palavra em $L$ aparece apenas uma vez na fita de saída de $M'$.

# Lema

Se $L$ é enumerada por uma máquina de Turing, então $L$ é recursivamente enumerável.

# Teorema

Uma linguagem é recursivamente enumerável se, e somente se, pode ser enumerada por uma máquina de Turing.

# Licença

- Slides feitos em LaTeX usando beamer e tikz, editados com vim.
- Licença

  Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.http://creativecommons.org/licenses/by-nc-nd/2.5/br/