

Linguagens Formais e Autômatos

Aula 21 - Variantes da Máquina de Turing

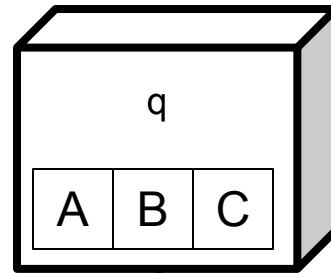
Referências bibliográficas

- **Introdução à teoria dos autômatos, linguagens e computação / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** ; tradução da 2.ed. original de Vandenberg D. de Souza. - Rio de Janeiro : Elsevier, 2002 (Tradução de: Introduction to automata theory, languages, and computation - ISBN 85-352-1072-5)
 - Capítulo 8 - Seções 8.3 e 8.4
- **Introdução à teoria da computação / Michael Sipser** ; tradução técnica Ruy José Guerra Barretto de Queiroz ; revisão técnica Newton José Vieira. -- São Paulo : Thomson Learning, 2007 (Título original : Introduction to the theory of computation. "Tradução da segunda edição norte-americana" - ISBN 978-85-221-0499-4)
 - Capítulo 3 - Seção 3.2

Variantes

- Existem muitas variantes para o modelo que vimos anteriormente
 - Todas possuem o mesmo poder de reconhecimento
- Facilitam o entendimento de alguns aspectos
- Facilitam a “programação”

Armazenamento no estado



E se eu quiser
armazenar informações
no controle finito?



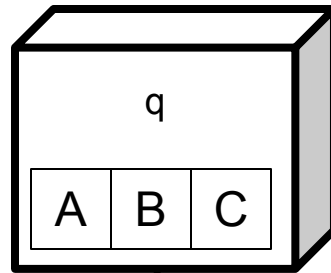
Armazenamento no estado

- Por exemplo, uma linguagem baseada em $\{a,b\}^*$ onde os primeiros três símbolos não se repetem ao longo da cadeia:
 - “aabbbabbbbaaa” faz parte (aab não se repete)
 - “ababbbaabbbabbb” faz parte (aba não se repete)
 - “aabbbbaabbbabba” não faz parte (aab aparece depois)
- Seria interessante “armazenar” as três primeiras letras e mantê-las “por perto” para poder consultar

Armazenamento no estado

- A solução é simples
- Não exige modificações no modelo da MT
 - Basta dar nomes adequados e criar quantos estados forem necessários para cobrir todas as combinações dos valores a serem armazenados
 - E também incluir os estados “normais” (q_1 , q_2 , etc)
- Ex:
 - $q=q_1, P_1=a, P_2=a, P_3=a \rightarrow q_1aaa$
 - $q=q_1, P_1=a, P_2=a, P_3=b \rightarrow q_1aab$
 - $q=q_4, P_1=a, P_2=b, P_3=a \rightarrow q_4aba$

Várias trilhas em uma única fita



E se eu quisesse armazenar mais de um valor em cada célula, ou dividir a fita em várias “trilhas”?

Trilha 1						X								
Trilha 2						Y								
Trilha 3						Z								

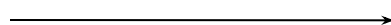
Várias trilhas em uma única fita

- Por exemplo, considere a seguinte linguagem:
 - $\{w#w \mid w \text{ em } \{0,1\}^*\}$
 - Anteriormente, projetamos uma MT que ziguezagueia pela entrada, “riscando” símbolos e substituindo-os por um x
 - Cada x representa um símbolo que já foi verificado
 - Quando toda a cadeia estiver somente com “x”s, ela é aceita
- Mas...
 - E se eu precisar de algum comportamento adicional?
 - E se eu precisar lembrar quais eram os símbolos que estavam lá?

Várias trilhas em uma única fita



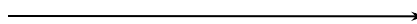
Trilha 1 (entrada)	0	0	1	0	1	#	0	0	1	0	1	
Trilha 2 (marcação)	X											



Trilha 1 (entrada)	0	0	1	0	1	#	0	0	1	0	1	
Trilha 2 (marcação)	X						X					



Trilha 1 (entrada)	0	0	1	0	1	#	0	0	1	0	1	
Trilha 2 (marcação)	X	X					X					

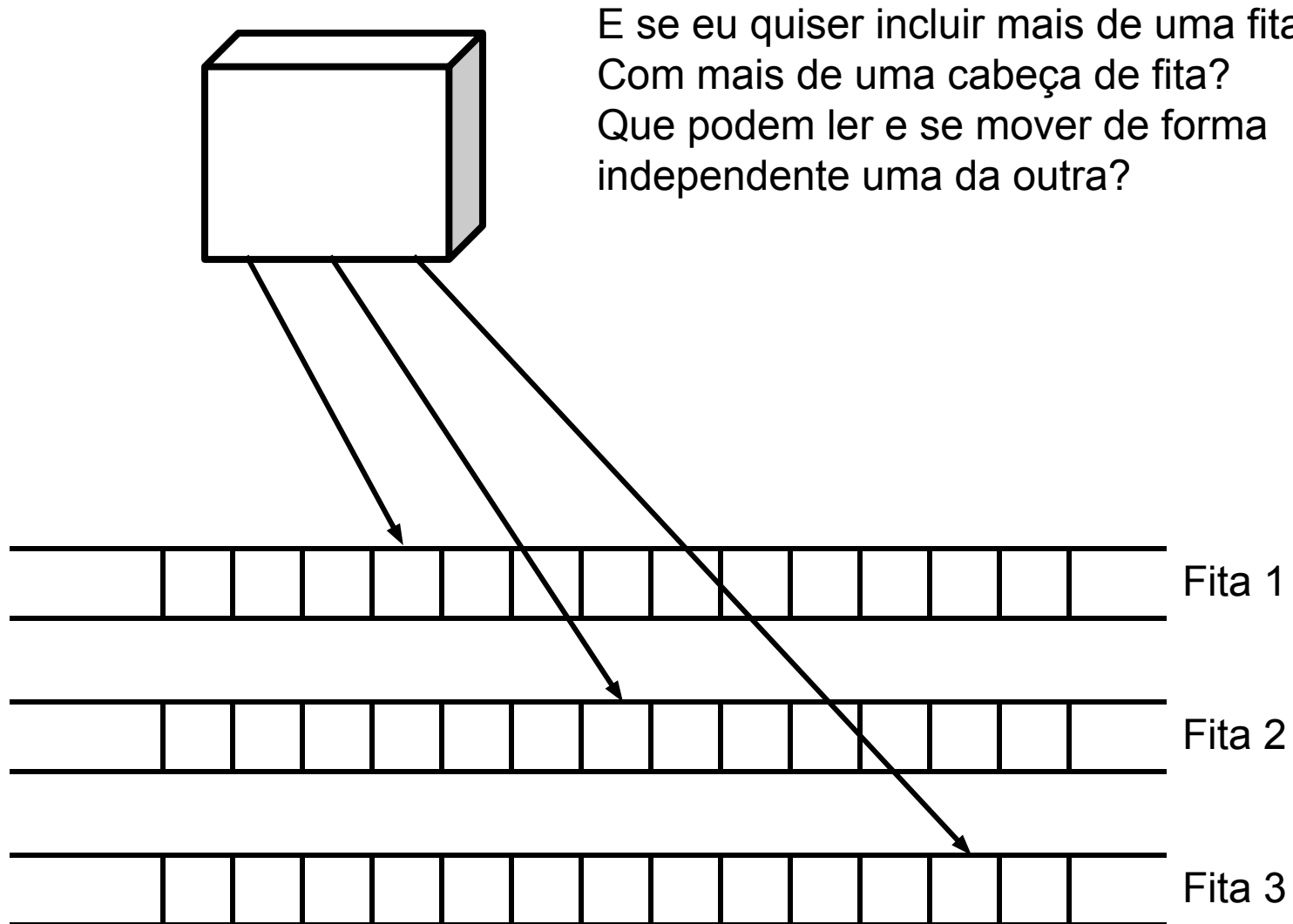


Trilha 1 (entrada)	0	0	1	0	1	#	0	0	1	0	1	
Trilha 2 (marcação)	X	X					X	X				

Várias trilhas em uma única fita

- O “truque” é fazer o mesmo que o armazenamento no estado
 - Ou seja, cada símbolo de fita, na verdade, é uma tupla que representa uma combinação diferente de símbolos, um para cada trilha
 - Por exemplo, na MT original (com uma única trilha):
 - $\Sigma = \{0,1\}$
 - $\Gamma = \{B,0,1,X\}$
 - Utilizando duas trilhas
 - $\Sigma = \{0,1\}$
 - $\Gamma = \{B,0,1,0',1'\}$
 - Onde 0' representa o 0 “marcado” com um X, e 1' representa o 1 “marcado” com um X
 - Ou: $\Gamma = \{(B,B),(0,B),(1,B),(0,X),(1,X)\}$
- Nessa situação, ainda é possível ver qual é o símbolo original, mesmo sendo “marcado”
 - É inclusive possível limpar a trilha auxiliar após seu uso
 - E utilizá-la para outra coisa

MT com várias fitas



MT com várias fitas

- A inclusão de outras fitas não aumenta o poder de definição de linguagens ao modelo básico, com uma única fita
 - Mas facilita a sua “programação”
 - Além de outras demonstrações interessantes

MT com várias fitas

- Inicialmente:
 - A entrada, uma sequência finita de símbolos de entrada, é colocada na primeira fita
 - Todas as outras células de todas as fitas contêm brancos
 - O controle finito está no estado inicial
 - A cabeça da primeira fita está na extremidade esquerda da entrada
 - Todas as outras cabeças de fitas estão em alguma célula arbitrária (na verdade, tanto faz)

MT com várias fitas

- Num movimento:
 - O controle entra em um novo estado (que pode ser o mesmo que o anterior)
 - Em cada fita, um novo símbolo de fita é escrito na célula varrida (pode ser o mesmo que o anterior)
 - Cada uma das cabeças faz um movimento
 - Para a esquerda, direita, ou estacionário (já vimos que movimento estacionário pode ser simulado por um movimento à esquerda seguido por um movimento à direita)
- Ou seja:
 - $\delta(q, a_1, a_2, \dots, a_k) = (p, b_1, b_2, \dots, b_k, E, D, E, \dots, E)$

MT com várias fitas

- MTs com uma fita reconhecem as linguagens recursivamente enumeráveis
 - MTs com múltiplas fitas também reconhecem as linguagens RE (óbvio)
 - Mas elas reconhecem mais linguagens?

Hierarquia	Gramáticas	Linguagens	Autômato mínimo
------------	------------	------------	-----------------

**Máquinas de Turing
com múltiplas fitas**

Tipo-0	Recursivamente Enumeráveis	Recursivamente Enumeráveis	Máquinas de Turing
Tipo-1	?	?	?
Tipo-2	Livres de contexto	Livres de contexto	Autômatos de pilha
Tipo-3	Regulares (Expressões regulares)	Regulares	Autômatos finitos

Estaria a MT com múltiplas fitas aqui em cima?

MT com várias fitas

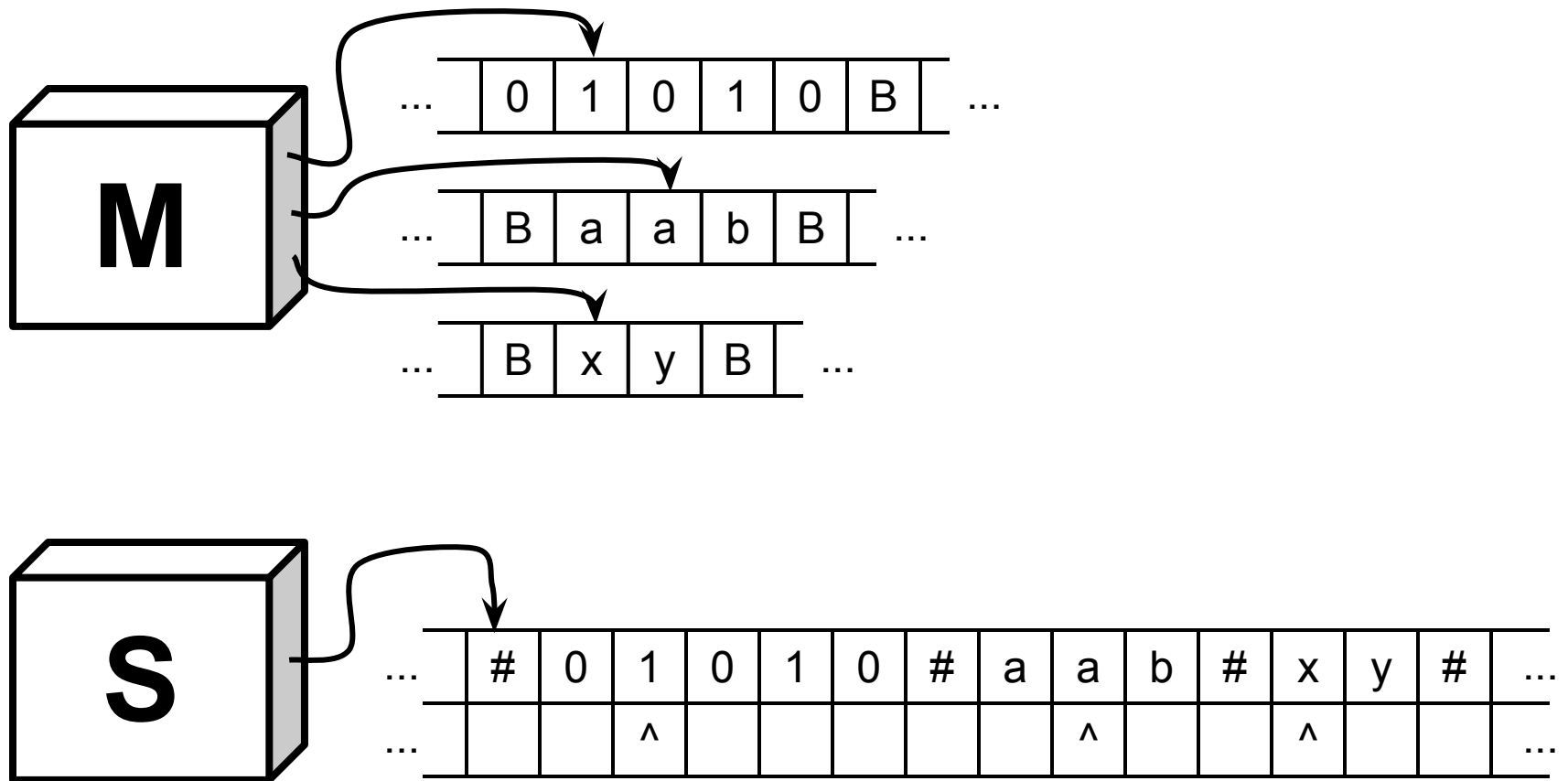
- A resposta é dada pelo seguinte teorema:

“Toda linguagem aceita por uma MT de várias fitas é recursivamente enumerável”

- Prova: por construção = conversão de múltiplas fitas para uma única fita
 - Prova 1
 - Prova 2

MT com várias fitas

- Prova 1: conversão de M para S



MT com várias fitas

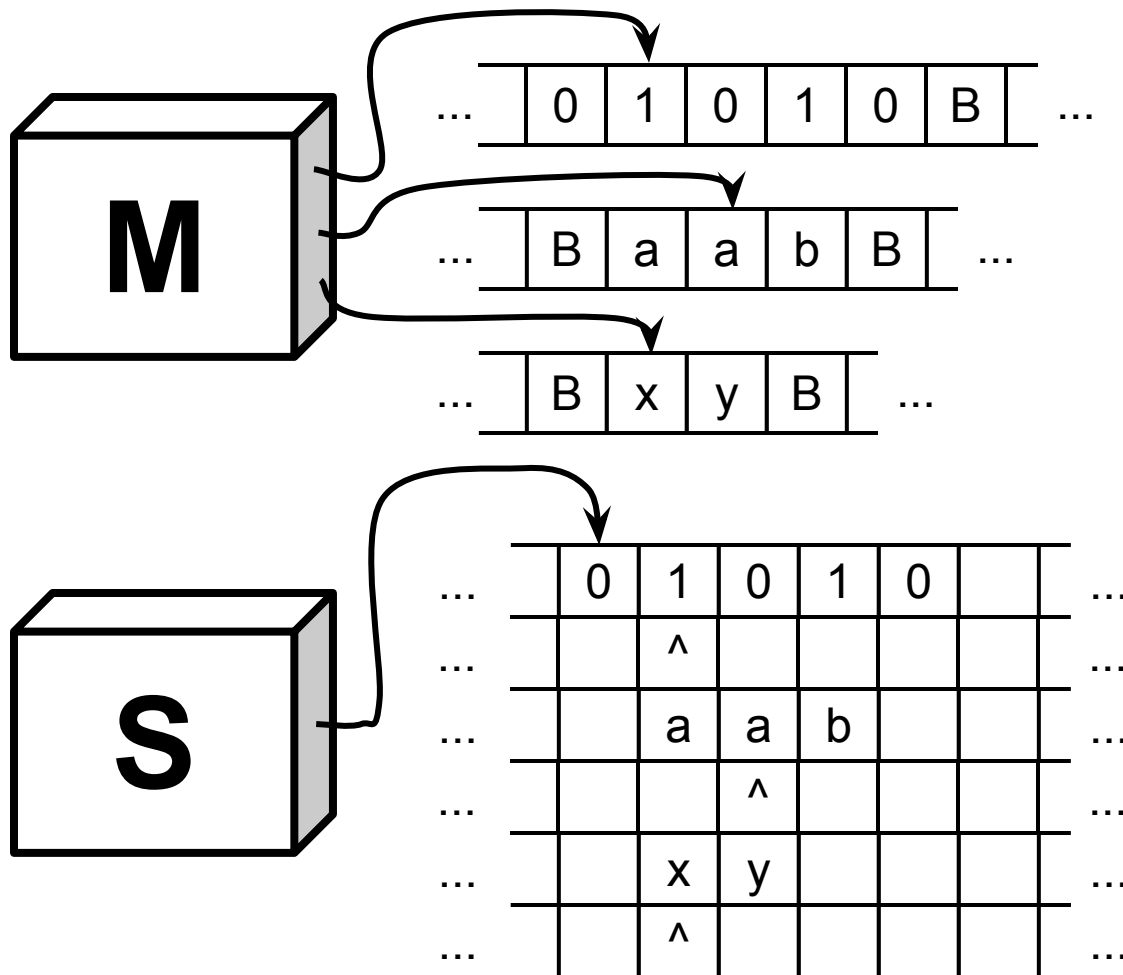
- Prova 1: conversão de M para S
 - M = múltiplas fitas
 - S = uma fita
- Inicialmente, a fita de S é formatada:
 - $\#w_1'w_2...w_n\#B'\#B'\#...\#$
 - O conteúdo de cada fita é colocado em sequência, separados por um símbolo especial (#)
 - A primeira trilha contém a cadeia de entrada, as demais contém brancos somente
 - Uma segunda trilha irá conter apenas “ponteiros” que apontam onde as cabeças estão em um determinado momento (representados por apóstrofos (') aqui)

MT com várias fitas

- Prova 1: conversão de M para S
- Em um único movimento:
 - S faz uma varredura na fita, desde o primeiro # até o último #
 - Nessa varredura, S “armazena” as posições de cada cabeça
 - S faz então uma segunda varredura, atualizando as fitas conforme a função de transição de M
- Se em algum ponto S move uma das cabeças virtuais sobre um #, significa que M teria encontrado um branco
 - S então escreve um branco sobre o # e desloca o restante da fita, símbolo a símbolo, aumentando seu comprimento, para a esquerda ou direita, conforme necessário

MT com várias fitas

- Prova 2: conversão de M para S



MT com várias fitas

- Prova 2: conversão de M para S
- Inicialmente:
 - Para cada fita em M, cria-se duas trilhas em S
 - Uma para os símbolos
 - Outra apenas para marcar a posição da cabeça de M
- Em um movimento:
 - S percorre as trilhas de marcação (armazenando quantos ela já descobriu, para não se perder)
 - E armazena os símbolos lidos na posição correspondente da trilha que representa a fita original
 - S volta ao início e percorre novamente as trilhas, atualizando os símbolos e as posições das cabeças virtuais, conforme definido em M

MT com várias fitas

- Um questionamento: por que não “armazenar” as posições das cabeças no controle finito?
 - Resposta: porque existem infinitas posições (as fitas são infinitas)
 - E o armazenamento no controle finito usa um número finito de estados
 - Ou seja, se eu estiver na posição 1456633 da fita, eu teria que ter um estado $q_{1456633}$
 - Mas sempre poderá existir uma posição 1456634!!
 - Ou seja, seria necessário ter infinitos estados

Uma análise importante

- MT com várias fitas e com uma fita são equivalentes:
 - em termos de capacidade reconhecedora !!
- Mas e em tempo de execução?
 - Obviamente gasta-se mais tempo executando uma MT com uma única fita!!
- De que isso importa?
 - Mais adiante usaremos MTs como uma forma de analisar tratabilidade!
 - Relacionado com tempos de execução viáveis ou inviáveis
 - Nesse contexto, a análise do tempo de execução faz toda a diferença!

Tempo de execução de uma MT

- Não é tempo real
 - Porque a MT não é uma máquina real!!
- É um tempo relativo:
 - Tempo de execução da MT M sobre entrada w é o número de movimentos que M faz antes de parar
 - Se M não parar em w , o tempo será infinito
- Isso leva ao estudo da complexidade
 - Veremos mais adiante na disciplina

Tempo de construção de muitas fitas para uma

- Aparentemente, a diferença de execução entre uma e várias fitas é grande
 - Mas não é muito!!
 - A MT de uma fita demora um tempo não maior que o quadrado do tempo tomado pela outra
 - Teorema: dada uma máquina de múltiplas fitas M que, dada uma entrada w , executa em n movimentos. Uma máquina de uma fita S equivalente a M , para a mesma entrada w , executa em $O(n^2)$ movimentos

Tempo de construção de muitas fitas para uma

- Prova: M (k fitas) convertido em S (1 fita)
- M executou n movimentos
 - Os marcadores de cabeça estarão com certeza distantes em $2n$ células ou menos
 - Ou seja, para cada um dos n movimentos de M :
 - S consegue, começando do marcador de cabeça mais à esquerda, em $2n$ movimentos no máximo, encontrar todos os marcadores nas suas respectivas trilhas
 - Em seguida, em $2n$ movimentos no máximo, S consegue com certeza voltar para o começo
 - Nesse caminho de volta, ele pode ter que ajustar as posições das cabeças conforme as transições de M
 - Ou seja, algumas cabeças irão para a direita, outras irão para a esquerda. Considerando k cabeças, em $2k$ movimentos ou menos é possível ajustar todas
 - Total até agora: $4n + 2k = O(n)$, pois k é uma constante
 - Se cada movimento de M leva $O(n)$, n movimentos em S levará n vezes isso, ou seja: $O(n^2)$

Tempo de construção de muitas fitas para uma

- Resumindo:
 - Se em múltiplas fitas leva 10 movimentos, em uma fita levará no máximo 100
 - Se em múltiplas fitas leva 100 movimentos, em uma fita levará no máximo 10000
- Essa diferença é pequena em termos computacionais!
- Pois é uma diferença polinomial
 - $y=x^2$
- O problema está em taxas de crescimento mais altas
 - São o limite entre o que podemos resolver por computadores e o que não tem solução prática
 - Mais detalhes depois

Fim

Aula 21 - Variantes da Máquina de Turing