ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Cap 3.2 – Variantes de MT

Introdução à Teoria da Computação. Michael Sipser. Thomson Learning, 2007.

Slides gentilmente cedidos pela
Profa. Ariane Machado Lima

Máquinas de Turing – Definição formal

Uma ***máquina de Turing*** é uma 7-upla, $(Q, \Sigma, \Gamma, \delta, q_0, q_{aceita}, q_{rejeita})$, 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. $\delta\colon Q \times \Gamma \longrightarrow Q \times \Gamma \times \{E, D\}$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial,
6. $q_{aceita} \in Q$ é o estado de aceitação, e
7. $q_{rejeita} \in Q$ é o estado de rejeição, onde $q_{rejeita} \neq q_{aceita}$.

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. $\delta: Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\mathrm{E}, \mathrm{D}\}$ 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 $q_{\text{rejeita}} \neq q_{\text{aceita}}$.

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

3.2 – Variantes de Máquinas de Turing

Variantes de Máquinas de Turing

Máquina de Turing é um modelo robusto: ela e suas variações reconhecem a mesma classe de linguagens

Robustez do Modelo da Máquina de Turing

- $M = (S, \Sigma, \Gamma, \delta, s_0, s_a, s_r)$.
  - $\delta : S \times \Gamma \to S \times \Gamma \times \{E, D\}$.
- $\delta : S \times \Gamma \to S \times \Gamma \times \{E, D, P\}$.
  - Permite a cabeça de leitura/gravação ficar parada.
  - Esta característica aumenta o poder do modelo?
  - Não! Pode-se simular um modelo com o outro.

MT com múltiplas fitas

- $M = (S, \Sigma, \Gamma^1, \Gamma^2, \ldots, \Gamma^k, \delta, s_0, s_a, s_r)$, onde:
  - $\delta : S \times \Gamma^k \rightarrow S \times \Gamma^k \times \{E, D\}^k$.
  - k : número de fitas.
- $\delta(s_i, a_1, \ldots, a_k) = (s_j, b_1, \ldots, b_k, E, D, \ldots, E)$:
  - s_i : estado corrente.
  - Cabeças 1 a k lêem símbolos $a_1, \ldots, a_k$.
  - s_j : próximo estado.
  - Cabeças 1 a k escrevem símbolos $b_1, \ldots, b_k$.

MT com múltiplas fitas

Teorema 2.18

Toda máquina de Turing com múltiplas fitas é equivalente a alguma máquina de Turing com apenas uma fita.

Demonstração.

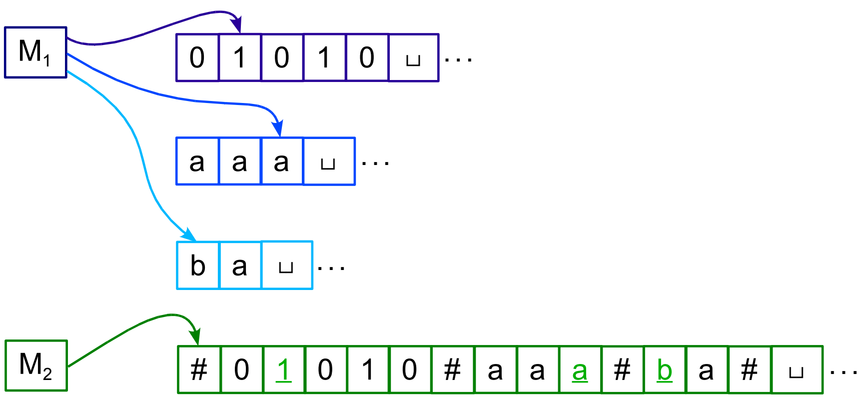
- Simular a máquina de Turing com k fitas com uma máquina de Turing com uma fita.
  - M_1 : Máquina com k fitas.
  - M_2 : Máquina com 1 fita.
  - Armazenar as k entradas da máquina M_1 na fita da máquina M_2.
  - Usar o símbolo # para separar as k entradas na fita M_2.
  - Marcar na fita da máquina M_2 as posições das cabeças de leitura/gravação.
    - Usar novos símbolos que devem ser adicionados ao alfabeto da fita.

□

MT com múltiplas fitas

Exemplo 2.19

- M_1 : Máquina com 3 fitas.
- M_2 : Máquina com 1 fita.



MT com múltiplas fitas

- Comportamento de M_2 com a entrada $w = w_1 \cdots w_n$:
  - Armazenar as k entradas da máquina M_1 na fita da máquina M_2: $\# \dot{w}_1 \, w_2 \cdots w_n \# \dot{\sqcup} \, \# \dot{\sqcup} \, \# \cdots \#$
  - Cabeça de leitura/gravação de M_2 move-se do primeiro # ao $(k + 1)$-ésimo # para determinar as posições da cabeças virtuais.
    - M_2 realiza segunda passagem na fita para atualização (conforme função de transição).
  - Se uma das cabeças virtuais de M_2 move-se para a direita e encontra um #:
    - M_1 move a correspondente cabeça para uma posição em branco da fita (ainda não lida).
    - M_2 desloca o conteúdo da fita (a partir do #) e escreve o símbolo $\sqcup$ na posição aberta na fita.

MT com múltiplas fitas

Recursivamente Enumerável

Corolário 2.20

Uma linguagem é Turing-reconhecível se e somente se alguma máquina de Turing com múltiplas fitas a reconhece.

Demonstração.

$\Rightarrow$

- Linguagem Turing-reconhecível é reconhecida por uma máquina de Turing com apenas uma fita.
- Máquina de Turing com apenas uma fita é um caso especial de máquina com múltiplas fitas.

$\Leftarrow$ Conseqüência do teorema anterior.

$\square$

Mostrar que:

1- toda mT com marcador de início de fita tem uma mT equivalente sem marcador de início fita

2- toda mT com fita limitada à esquerda tem uma mT equivalente com fita ilimitada em ambos os sentidos

Máquinas de Turing Não-Determinísticas

Definição importante, usada no conceito de problemas NP.

O que significa NP?

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

Conjunto das Partes: Power Set

A computação de uma maquina de Turing não determinística 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.

TEOREMA 3.16

Toda máquina de Turing não-determinística tem uma máquina de Turing determinística que lhe é equivalente.

TEOREMA 3.16

Toda máquina de Turing não-determinística tem uma máquina de Turing determinística que lhe é equivalente.

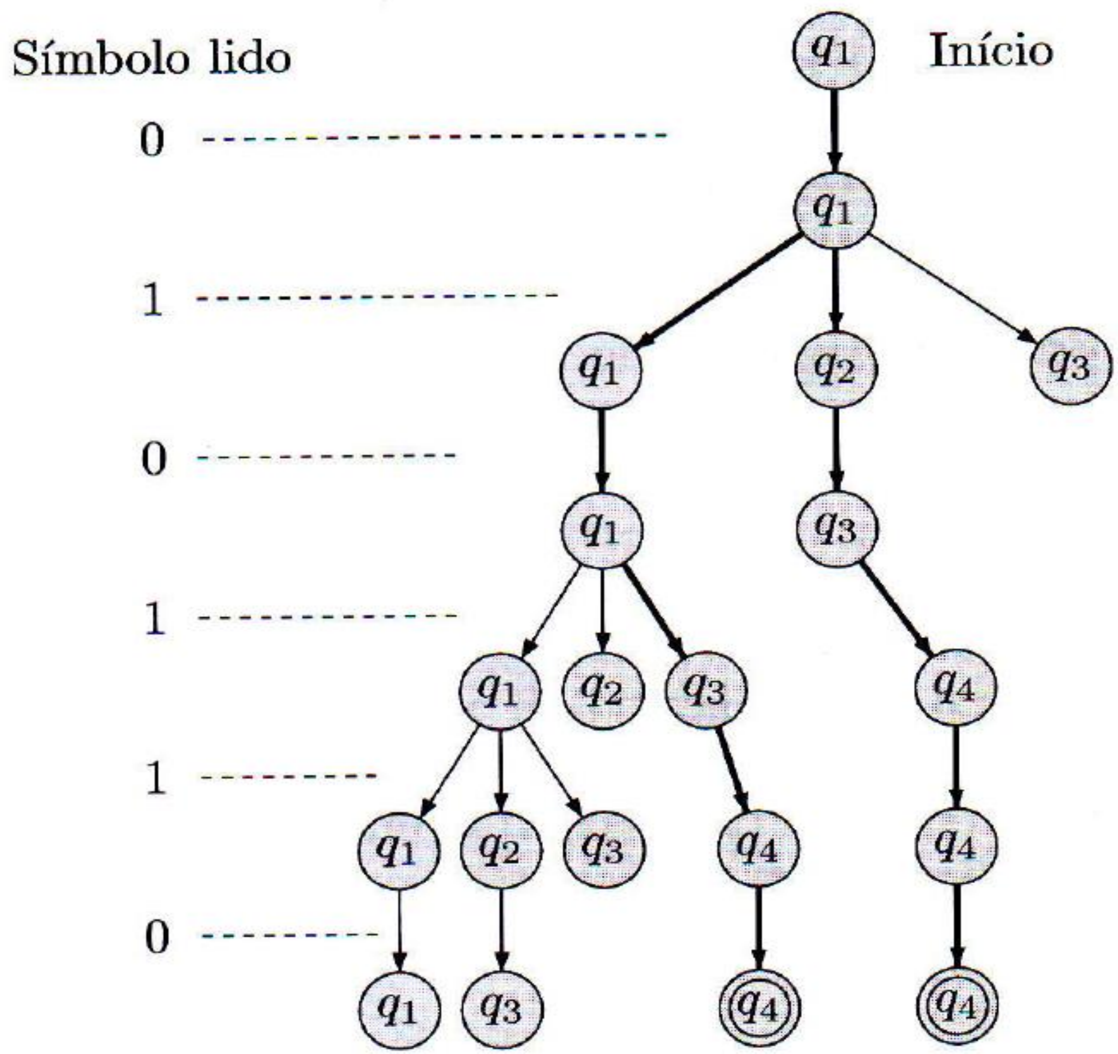
Ideia da prova:

Usar para MT determinística D para simular todas as possíveis computações de uma MT não determinística N

Computação de N representada por uma árvore (filhos de um nó são as possibilidades de transição)

Cada caminho (a partir da raiz) é uma computação possível

Lembram da árvore para AFNDs?



ir para o exemplo do Marcus

TEOREMA 3.16

Toda máquina de Turing não-determinística tem uma máquina de Turing determinística que lhe é equivalente.

Ideia da prova:

Usar para MT determinística D para simular todas as possíveis computações de uma MT não determinística N

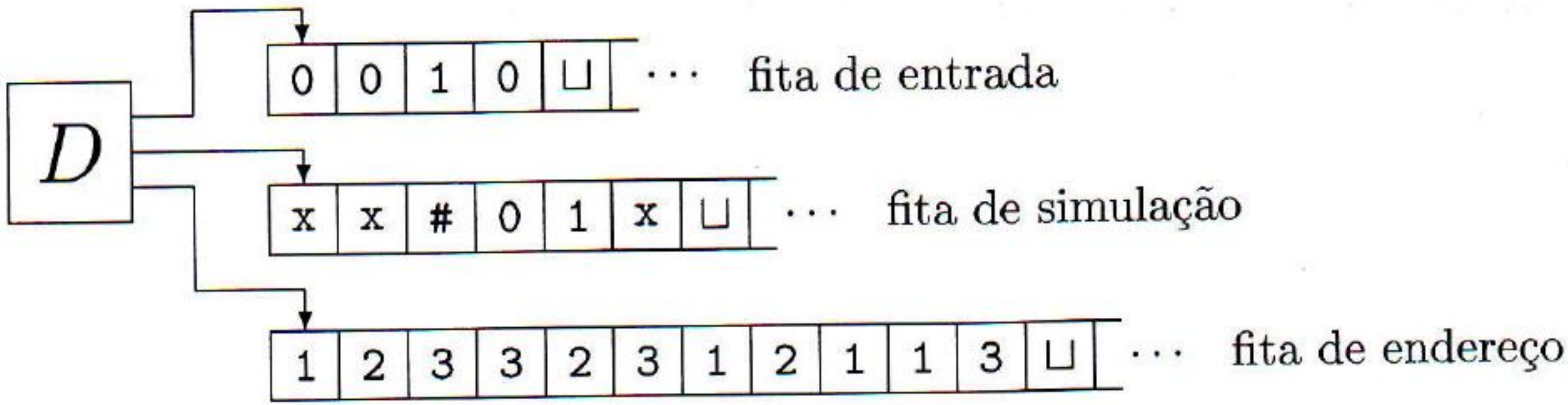
Computação de N representada por uma árvore (filhos de um nó são as possibilidades de transição)

Cada caminho (a partir da raiz) é uma computação possível

Cada nó dessa árvore terá um endereço que indica qual filho seguir a partir da raiz (ex: 314)

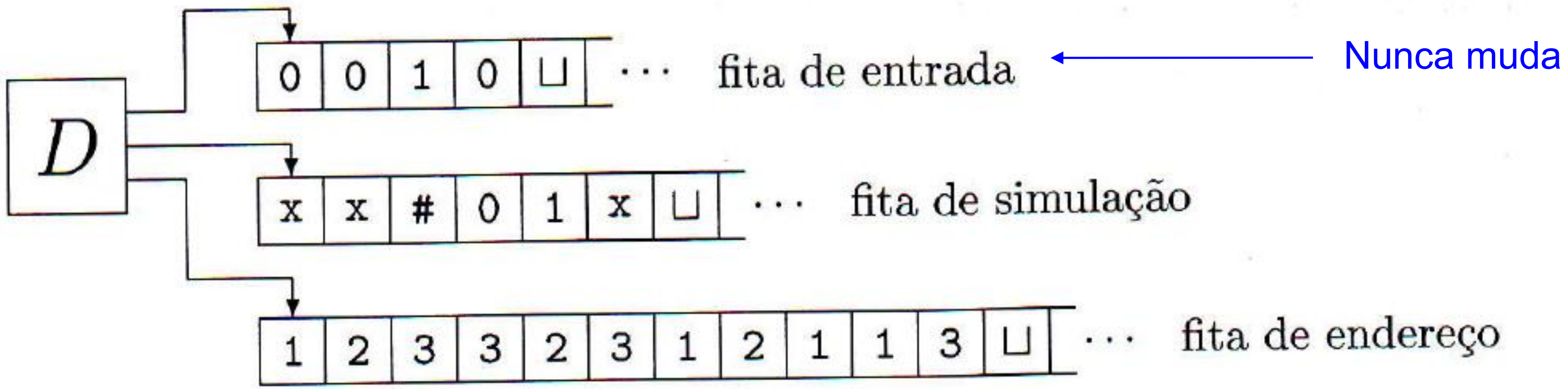
D irá percorrer essa árvore através de busca em largura (para impedir que caia em um ramo infinito)

Prova



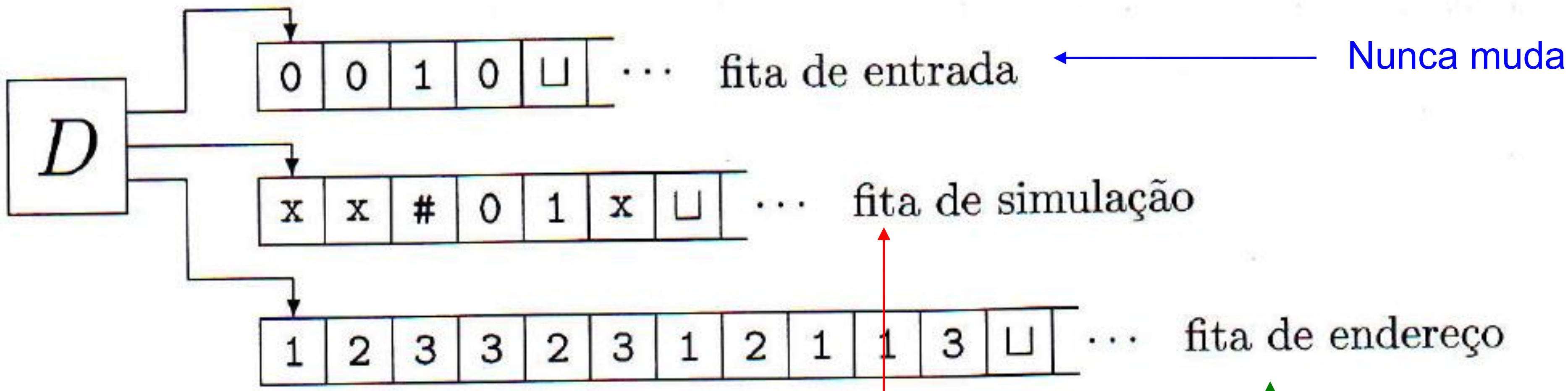
D
0
0
1
0
⊔
···
fita de entrada
x
x
#
0
1
x
⊔
···
fita de simulação
1
2
3
3
2
3
1
2
1
1
3
⊔
···
fita de endereço

Prova



D
0 0 1 0 ⊔ ···
fita de entrada
Nunca muda
x x # 0 1 x ⊔ ···
fita de simulação
1 2 3 3 2 3 1 2 1 1 3 ⊔ ···
fita de endereço

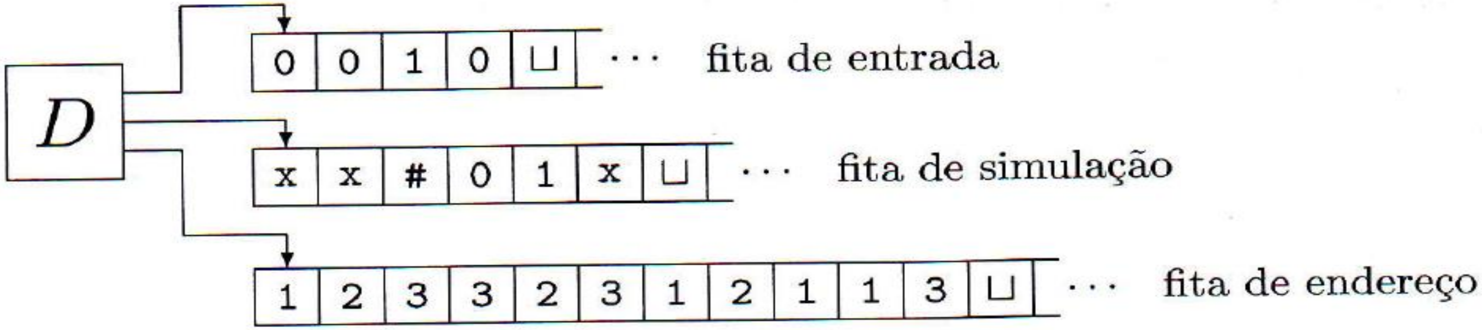
Prova



Usada para simular cada ramo separadamente

Usada para indicar em qual computação (caminho) estou
Possui uma cadeia em Σ_b^* , $\Sigma_b = \{1, 2, \ldots, b\}$, sendo b o maior número de filhos de um nó (ie, o maior número de transições alternativas de um estado)

Prova



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 simular N com a entrada w sobre um ramo de sua computação não-determinística. Antes de cada passo de N, consulte o próximo símbolo na fita 3 para determinar qual escolha fazer entre aquelas permitidas pela função de transição de N. Se não restam mais símbolos na fita 3 ou se essa escolha não-determinística for inválida, aborte esse ramo indo para o estágio 4. Também vá para o estágio 4 se uma configuração de rejeição for encontrada. Se uma configuração de aceitação for encontrada, *aceite* a entrada.
4. Substitua a cadeia na fita 3 pela próxima cadeia na ordem lexicográfica. Simule o próximo ramo da computação de N indo para o estágio 2.

COROLÁRIO 3.18

Uma linguagem é Turing-reconhecível se e somente se alguma máquina de Turing não-determinística a reconhece.

MT não determinística ($N\cancel{D}TM$)

- $M = (S, \Sigma, \Gamma, \delta, s_0, s_a, s_r)$, onde:
  - $\delta : S \times \Gamma \rightarrow \mathcal{P}(S \times \Gamma \times \{E, D\})$.
- Processamento de uma NTM:
  - Subconjuntos de estados e transições organizados como uma árvore.
  - Ramos correspondem a diferentes opções de processamentos da máquina (não determinismo).
  - Se algum ramo leva a um estado de aceitação, a máquina aceita a entrada.

MT não determinística ($N\cancel{D}TM$)

Teorema 2.21

Toda máquina de Turing não determinística (NTM) é equivalente a alguma máquina de Turing determinística (DTM).

Esquema da prova

- Simular uma NTM qualquer com uma DTM.
- DTM deve testar todos os ramos do processamento da NTM.
- Encontrou um estado de aceitação em algum dos ramos?

  Sim DTM `aceita`.

  Não Simulação não termina.

Lembrete:
Duas máquinas são equivalentes se elas reconhecem a mesma linguagem

Resultados possíveis de uma mt M:
*** aceita, rejeita ou cicla.
*** M falha em aceitar uma cadeia se entra no estado sr (rejeição) ou quando cicla (entra em loop infinito)

MT não determinística ($N\cancel{D}TM$)

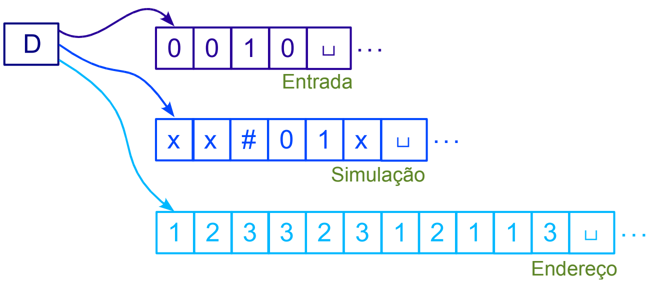
Esquema da prova

- NTM N e cadeia w.
- Processamento de w é uma árvore T:
  - Cada ramo é uma das opções do não determinismo.
  - Cada nó é uma configuração de N.
  - Raiz de T é a configuração inicial.
- Busca em profundidade em T, por DTM D, de um estado de aceitação:
  - D pode descer em um ramo infinito e nunca encontrar estado de aceitação em outro ramo.
- Busca em largura em T:
  - Garantia de que D visita cada nó de T até encontrar configuração de aceitação.

MT não determinística ($N\cancel{D}TM$)

Esquema da prova

- DTM com três fitas:
  - Fita 1 → cadeia de entrada (nunca é alterada).
  - Fita 2 → copia da fita da NTM em algum ramo do processamento.
  - Fita 3 → posição da DTM na árvore da processamento da NTM.



MT não determinística ($N\cancel{D}TM$)

Esquema da prova

- Fita 3 da DTM:
  - Cada nó da árvore tem no máximo b filhos.
    - $b \rightarrow$ tamanho do maior conjunto possível de escolhas numa transição da NTM.
  - Cada nó recebe um endereço sobre o alfabeto $\Sigma_b = \{1, 2, \ldots, b\}$.
    - $231 \rightarrow 2^o$ filho da raiz, 3^o filho deste 2^o e 1^o filho desse 3^o.
  - Cada símbolo indica a próxima transição da NTM a ser simulada.
  - Endereço pode ser inválido e não corresponder a um nó.
    - Símbolo não corresponde a uma escolha disponível para a configuração.
  - Cadeia sobre Σ_b representa ramo do processamento da NTM <u>da raiz até o nó endereçado pela cadeia.</u>

cadeia vazia é o endereço da raiz da árvore

MT não determinística ($N\cancel{D}TM$)

Esquema da prova

- Funcionamento da DTM:
  1. Fita 1 contém cadeia w e fitas 2 e 3 estão vazias.
  2. Copiar Fita 1 para Fita 2.
  3. Usar Fita 2 para simular a NTM com entrada w em um ramo do processamento.
     - Antes de cada passo da NTM, consultar o próximo símbolo na Fita 3 para determinar a transição a ser simulada.
     - Se Fita 3 está vazia ou escolha é inválida, abortar esse ramo e ir para o passo 4.
     - Se encontrar configuração de rejeição, ir para o passo 4.
     - Se encontrar configuração de aceitação, aceitar a cadeia w.
  4. Trocar a cadeia da Fita 3 (próxima cadeia na ordem lexicográfica). Voltar ao passo 2 (Simular o processamento da NTM no próximo ramo). não há backtracking, volta para a raiz da árvore

Endereços em ordem lexicográfica (b=3):

e Como seria a busca na árvore
1 de computação?
2
3
11
12
13
21
22
23
31
32
33 111 112 ...

MT não determinística ($N\cancel{D}TM$)

Corolário 2.22

Uma linguagem é Turing-reconhecível se e somente se alguma máquina de Turing não determinística a reconhece.

Demonstração.

$\Rightarrow$ Qualquer DTM é automaticamente uma NTM.

$\Leftarrow$ Conseqüência do Teorema anterior.

□

MT não determinística ($NDTM$)

Corolário 2.23

Uma linguagem é decidível se e somente se alguma máquina de Turing não determinística a decide.

Demonstração.

- Modificar a prova do teorema anterior para a DTM parar em um ramo do processamento no qual NTM também pára.

□

Um bom exercício!

Atividade:
1. Como calcular a complexidade de tempo de mTs?
2. Como calcular a complexidade de tempo da NT M?
3. Vamos supor que a complexidade de NT M seja T(n), onde lwl = n.
qual seria complexidade de tempo da DT M?
4. Qual a relação entre mT não deterministas e
o estudo de classes de complexidade?
Sempre justificar suas respostas.

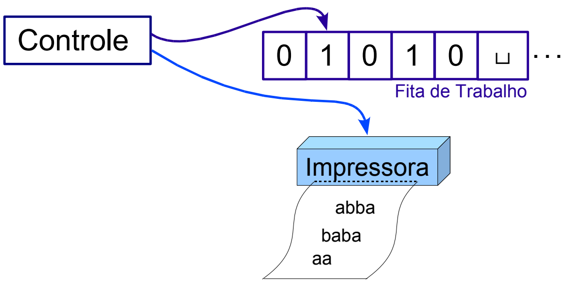
Decisor

- Máquina de Turing não determinística que pára em todos os ramos do processamento para todas as cadeias de entrada.

reconhecem (decidem) Linguagens Turing Decidíveis

Enumerador (Contador)

- Máquina de Turing com uma impressora acoplada.
- Dispositivo de saída para cadeias "geradas" pela máquina.



Enumerador (Contador)

- Começa o processamento com a fita de trabalho vazia.
- Se não pára, pode imprimir uma lista infinita de cadeias.
- A linguagem enumerada por E é a coleção de cadeias eventualmente impressas.
- E pode gerar as cadeias da linguagem em qualquer ordem, com possíveis repetições.

Enumerador (Contador)

Teorema 2.24

Uma linguagem é Turing-reconhecível se e somente se algum Enumerador a gera.

Demonstração.

$\Longleftarrow$

- Enumerador E que gera a linguagem $\mathcal{L}$.
- Máquina de Turing M que reconhece $\mathcal{L}$.
- Processamento de M com a cadeia $w \in \mathcal{L}$: Sigma*
  1. Execute E. Compare w com cada cadeia gerada por E.
  2. Se w aparece na saída de E, aceite.

□

Enumerador (Contador)

Teorema 2.24

Uma linguagem é Turing-reconhecível se e somente se algum Enumerador a gera.

Demonstração.

$\Longrightarrow$

- Máquina de Turing M que reconhece $\mathcal{L}$.
- Enumerador E que gera a linguagem $\mathcal{L}$.
- $s_1, s_2, \ldots, s_i \mapsto$ lista de todas as possíveis cadeias em Σ^*.
- Processamento de E:
  1. Ignore a cadeia de entrada.
  2. Repita os passos 3 e 4 para $i = 1, 2, 3, \ldots$.
  3. Execute M por i passos para cada $s_1, s_2, \ldots, s_i$.
  4. Se qualquer processamento aceita, imprima a correspondente cadeia s.

$\square$

Esse procedimento dá o efeito de se rodar M em paralelo sobre todas as possíveis cadeias de entrada

Enumerador (Contador)

Teorema 2.24

Uma linguagem é Turing-reconhecível se e somente se algum Enumerador a gera.

Demonstração.

$\Longrightarrow$

- Se M aceita uma cadeia w, eventualmente w vai aparecer na lista gerada por E.
  - Aparece na lista infinitas vezes.
  - M é executada desde o início para cada cadeia e repetição do passo 2.
- Simula a execução de M em paralelo para todas as possíveis cadeias de entrada.

□

Uma linguagem L é recursivamente enumerável (Turing-reconhecível) sse ~~se~~ existe uma mT M que enumera as cadeias de L.
M pode gerar as cadeias da linguagem em qualquer ordem, com possíveis repetições.
Neste caso, a mT M é considerada um procedimento gerador.

Um exemplo bem didático de mT não determinística
dos slides do professor Marcus Ramos

Não-determinismo

Definição

Uma Máquina de Turing M é dita "não-determinística" se existir mais de uma possibilidade de movimentação a partir de uma mesma configuração. Formalmente:

$$\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$$

Não-determinismo

Linguagem definida

Seja M uma Máquina de Turing M não-determinística e $w \in \Sigma^*$. São considerados três casos, que cobrem todas as situações possíveis:

- $w \in ACEITA(M)$ se e somente se existe pelo menos uma seqüência de movimentos que conduz M a um estado final com a cadeia w;
- $w \in REJEITA(M)$ se e somente se todas as seqüências de movimentos de M com a cadeia w conduzem à configurações de parada não-finais;
- $w \in LOOP(M)$ se e somente se:
  - Não existe nenhuma seqüência de movimentos que conduza M a um estado final com a cadeia w;
  - Existe pelo menos uma seqüência de movimentos que fazem com que M entre em loop com a cadeia w.

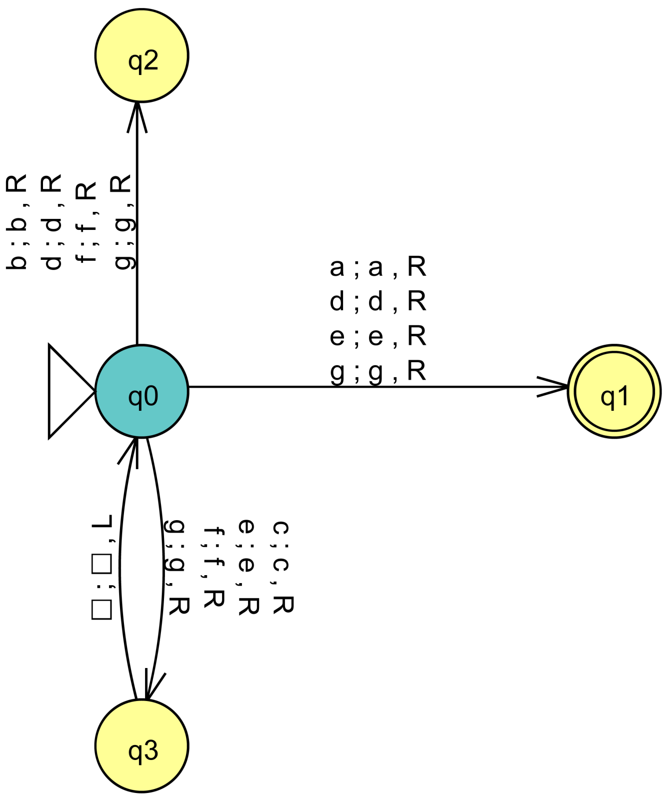
Não-determinismo

Exemplo

- A Máquina de Turing da figura seguinte é não-determinística e possui $\Sigma = \{a, b, c, d, e, f, g\}$;
- São consideradas cadeias de entrada que provocam todas as combinações possíveis entre as situações de aceitação, rejeição e loop, inclusive combinações duas a duas e as três simultaneamente;
- O resultado serve para ilustrar a determinação de $ACEITA(M), REJEITA(M)$ e $LOOP(M)$ em Máquinas de Turing não-determinísticas.

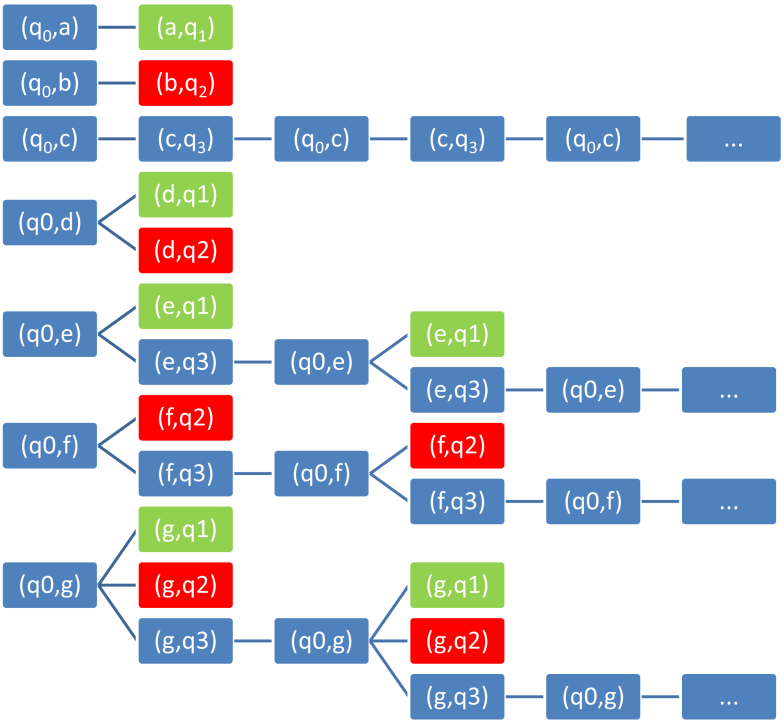
Não-determinismo

Exemplo



Não-determinismo

Exemplo



Não-determinismo

Exemplo

	ACEITA	REJEITA	LOOP	
a	✓			ACEITA
b		✓		REJEITA
c			✓	LOOP
d	✓	✓		ACEITA
e	✓		✓	ACEITA
f		✓	✓	LOOP
g	✓	✓	✓	ACEITA

Não-determinismo

Exemplo

Portanto, M particiona Σ^* nos seguintes conjuntos:

- $ACEITA(M) = \{a, d, e, g, ...\}$;
- $REJEITA(M) = \{b, ...\}$;
- $LOOP(M) = \{c, f, ...\}$;