ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 17
Máquinas de Turing e linguagens sensíveis ao contexto

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

Aula anterior

Máquinas de Turing - Tipos de descrições

1-) Descrição formal: diagrama de estados ou definição matemática (conjuntos, função de transição, etc.)

2-) Descrição de implementação (descrição em língua natural da manipulação da máquina – como escreve/lê da fita)

3-) Descrição de alto-nível (algoritmo - pseudocódigo)

Máquinas de Turing – Definição formal

Máquina pára IMEDIATAMENTE

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

Mais precisamente:

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

Máquinas de Turing – Definição formal

Nesta definição ela é determinística ou não determinística?

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

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

Máquinas de Turing – Definição formal

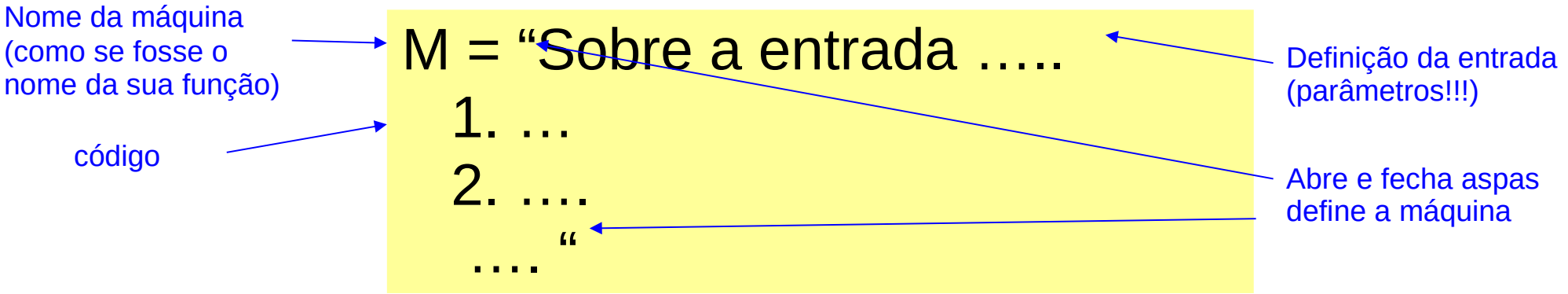
Nesta definição ela é determinística ou não determinística? **DETERMINÍSTICA!!!**

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

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

Máquinas de Turing - Tipos de descrições



2-) Descrição de implementação (descrição em língua natural da manipulação da máquina)

3-) Descrição de alto-nível (algoritmo)

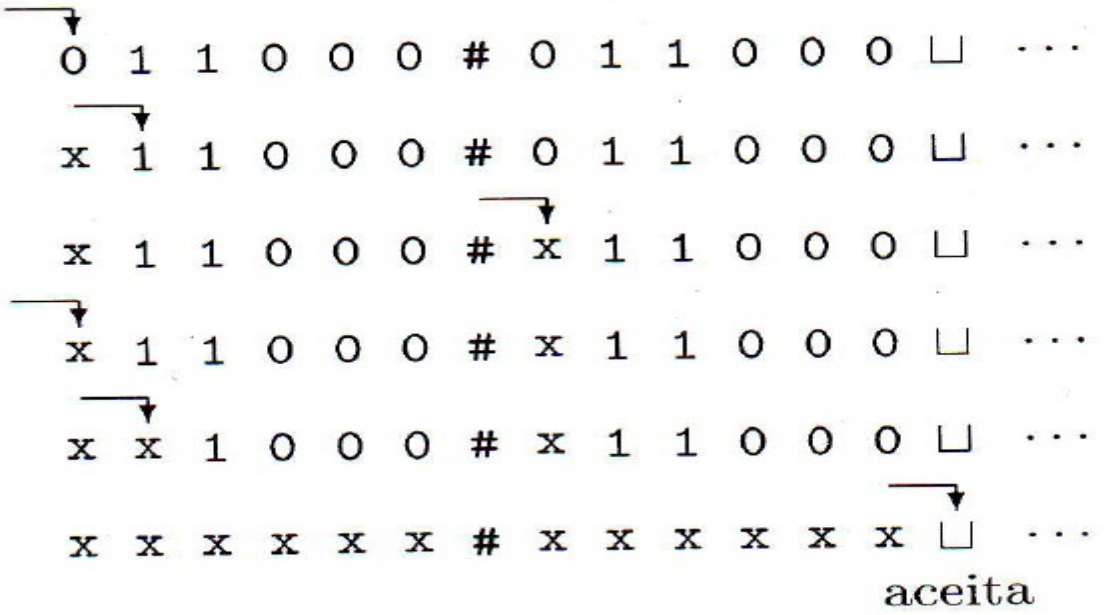
São como funções booleanas:

```
bool M(...){
  …
}
```

que dentro do código precisam retornar *true* ou *false* dependendo da entrada
Isto é: *aceitar* ou *rejeitar* a entrada

Máquinas de Turing - Exemplo

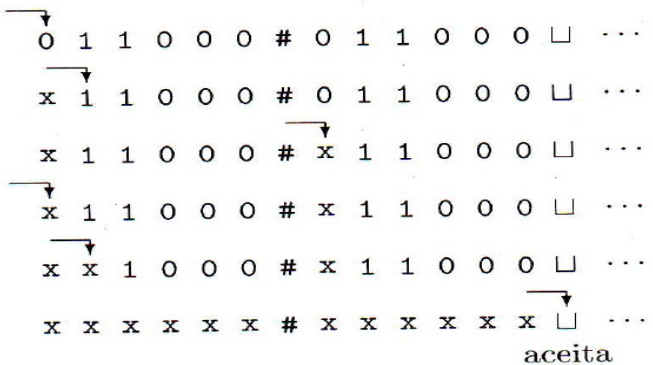
B= { w # w | w pertence a {0,1}*}

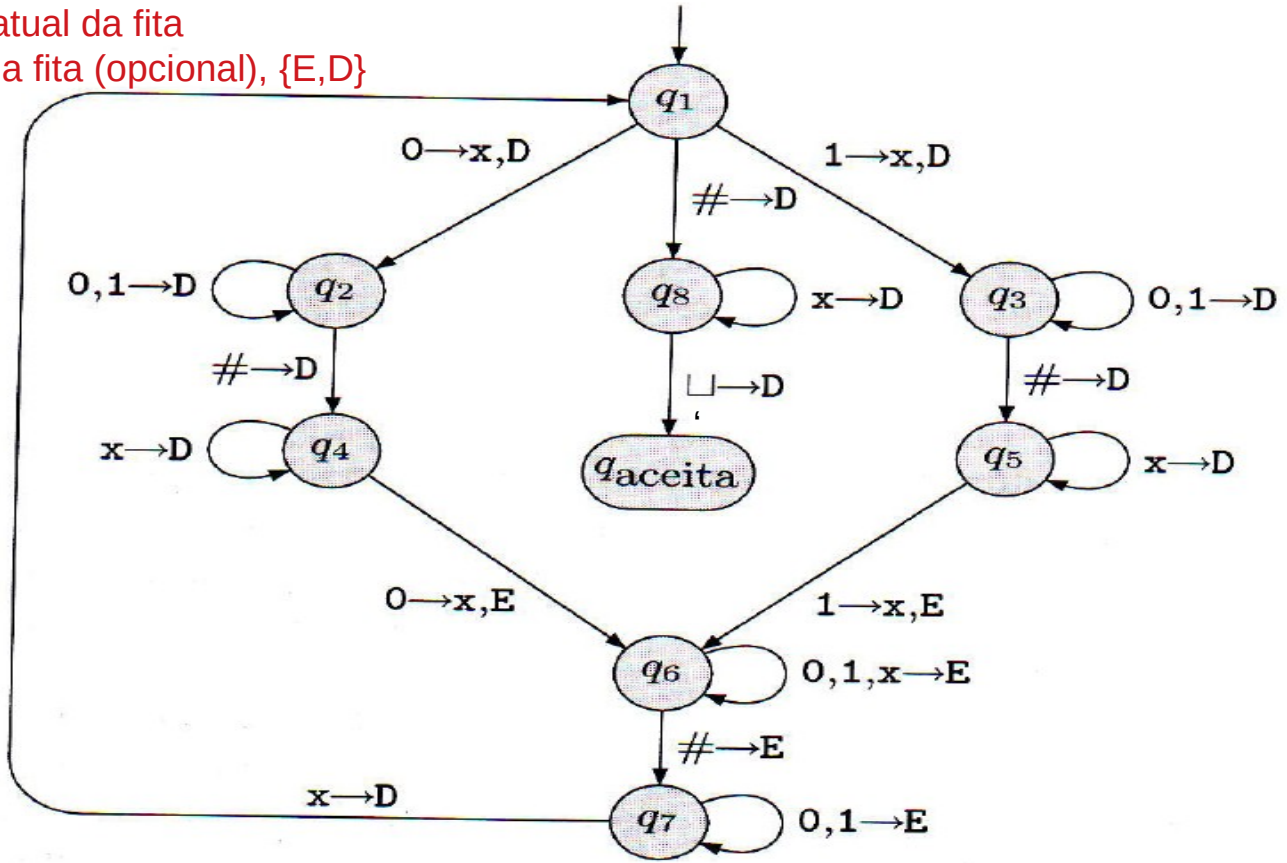


Notação: A → B:
A: lista de possíveis símbolos na posição atual da fita
B: símbolo a ser escrito na posição atual da fita (opcional), {E,D}

Descrição formal

Cada laço tenta casar um símbolo do lado esquerdo do # com um símbolo do lado direito, até que todas as correspondências sejam encontradas





Transições implícitas para $q_{rejeita}$ (indo para a direita, por convenção) quando aparece um símbolo não definido na transição.

Máquinas de Turing – Exemplo (descrição de implementação)

B= { w # w | w pertence a {0,1}*}

M_1 = "Sobre a cadeia de entrada w:

1. Faça um zigue-zague ao longo da fita checando posições correspondentes de ambos os lados do símbolo # para verificar se elas contêm o mesmo símbolo. Se elas não contêm, ou se nenhum # for encontrado, *rejeite*. Marque os símbolos à medida que eles são verificados para manter registro de quais símbolos têm correspondência.
2. Quando todos os símbolos à esquerda do # tiverem sido marcados, verifique a existência de algum símbolo remanescente à direita do #. Se resta algum símbolo, *rejeite*; caso contrário, *aceite*."

Máquinas de Turing – Exemplo (descrição de alto-nível)

B= { w # w | w pertence a {0,1}*}

M1 = “Sobre a cadeia de entrada w:

1. Verifique se todo o lado esquerdo de “#” é exatamente igual ao lado direito.
2. Se forem iguais *aceite*, senão *rejeite*.”

Máquinas de Turing

A coleção de cadeias que M aceita é ***a linguagem de M***, ou ***a linguagem reconhecida por M***, denotada $L(M)$.

DEFINIÇÃO 3.5

Chame uma linguagem de ***Turing-reconhecível***, se alguma máquina de Turing a reconhece.[1]

1 - Ou linguagem **recursivamente enumerável** ou linguagem **irrestrita**

A máquina de Turing pode NÃO PARAR para algumas cadeias (neste caso a cadeia não pertence à linguagem reconhecida pela máquina)

Máquinas de Turing (MT) Decisoras

Uma MT é decisora se ela nunca entra em loop (isto é, sempre pára em um estado de aceitação ou de rejeição).

Dizemos que um decisor que reconhece uma linguagem decide essa linguagem.

DEFINIÇÃO 3.6

Chame uma linguagem de ***Turing-decidível*** ou simplesmente ***decidível*** se alguma máquina de Turing a decide.[2]

2 - Ou linguagem **recursiva**

Aula de hoje

- Mais um exemplo de MT
- MT de fita limitada (Linguagens sensíveis ao contexto)
- Variantes de Máquinas de Turing

Máquinas de Turing - Exemplos

EXEMPLO 3.7

Aqui descrevemos uma máquina de Turing (MT) M_2 que decide $A = \{0^{2^n} \mid n \geq 0\}$, a linguagem consistindo em todas as cadeias de 0s cujo comprimento é uma potência de 2.

Máquinas de Turing
Decrição de alto-nível

EXEMPLO 3.7

Aqui descrevemos uma máquina de Turing (MT) M_2 que decide $A = \{0^{2^n} \mid n \geq 0\}$, a linguagem consistindo em todas as cadeias de 0s cujo comprimento é uma potência de 2.

M2 = “Sobre a cadeia de entrada w,

1. Conte o número de zeros.
2. Se o número de zeros for uma potência de 2, *aceite*. Caso contrário, *rejeite*.”

Máquinas de Turing
Descrição em nível de implementação

EXEMPLO 3.7

Aqui descrevemos uma máquina de Turing (MT) M_2 que decide $A = \{0^{2^n} \mid n \geq 0\}$, a linguagem consistindo em todas as cadeias de 0s cujo comprimento é uma potência de 2.

- Ideia: Uma potência de 2, sempre que eu divido por 2, terei outra potência de dois que é um número par ou o número 1

Máquinas de Turing
Descrição em nível de implementação

EXEMPLO 3.7

Aqui descrevemos uma máquina de Turing (MT) M_2 que decide $A = \{0^{2^n} \mid n \geq 0\}$, a linguagem consistindo em todas as cadeias de 0s cujo comprimento é uma potência de 2.

M_2 = "Sobre a cadeia de entrada w:

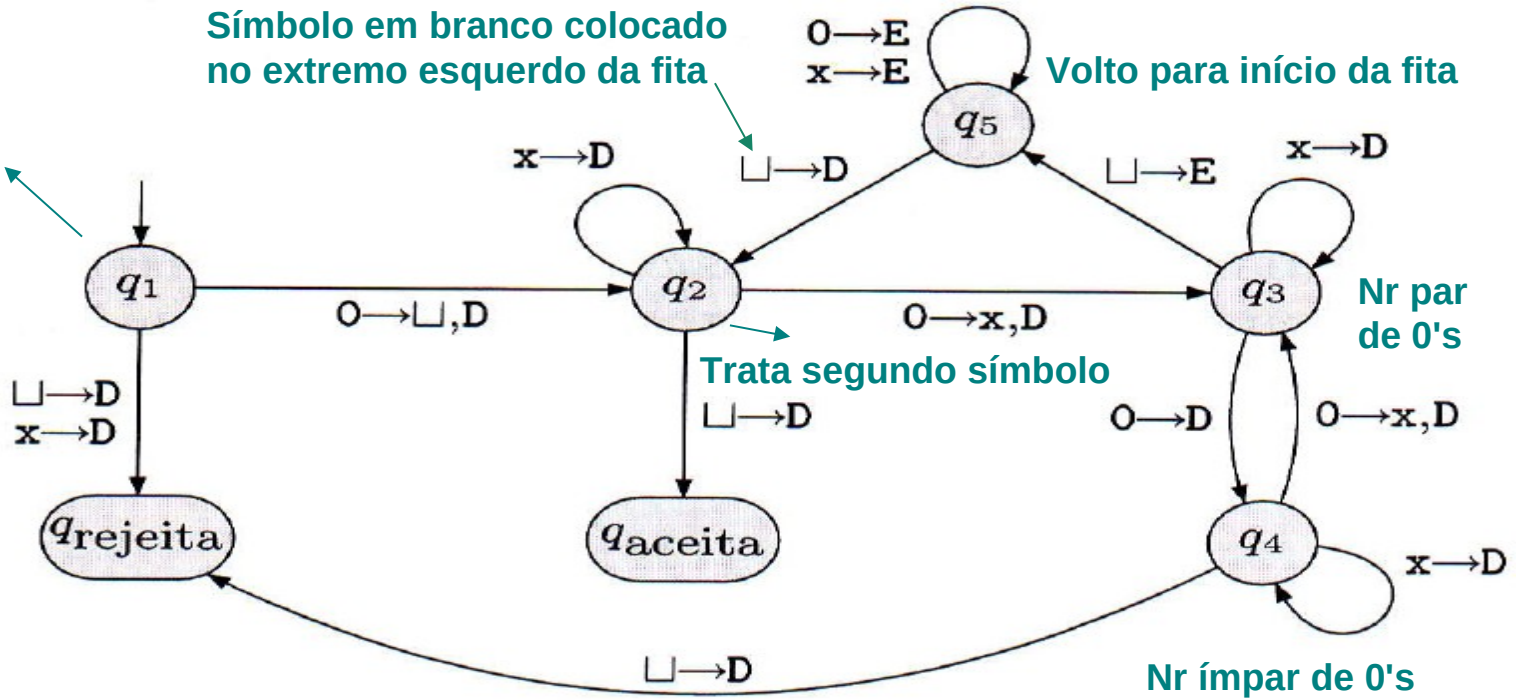
1. Faça uma varredura da esquerda para a direita na fita, marcando um 0 não, e outro, sim.
2. Se no estágio 1, a fita continha um único 0, *aceite*.
3. Se no estágio 1, a fita continha mais que um único 0 e o número de 0s era ímpar, *rejeite*.
4. Retorne a cabeça para a extremidade esquerda da fita.
5. Vá para o estágio 1."

Máquinas de Turing
Descrição formal

Agora, damos a descrição formal de $M_2 = (Q, \Sigma, \Gamma, \delta, q_1, q_{aceita}, q_{rejeita})$:

- $Q = \{q_1, q_2, q_3, q_4, q_5, q_{aceita}, q_{rejeita}\}$,
- $\Sigma = \{0\}$ e
- $\Gamma = \{0, x, \sqcup\}$.
- Descrevemos δ com um diagrama de estados (veja a Figura 3.8).
- Os estados inicial, de aceitação e de rejeição são q_1, q_{aceita} e $q_{rejeita}$.

Trata primeiro símbolo na primeira iteração (símbolo em branco colocado no lugar do primeiro 0 para indicar o início da fita)



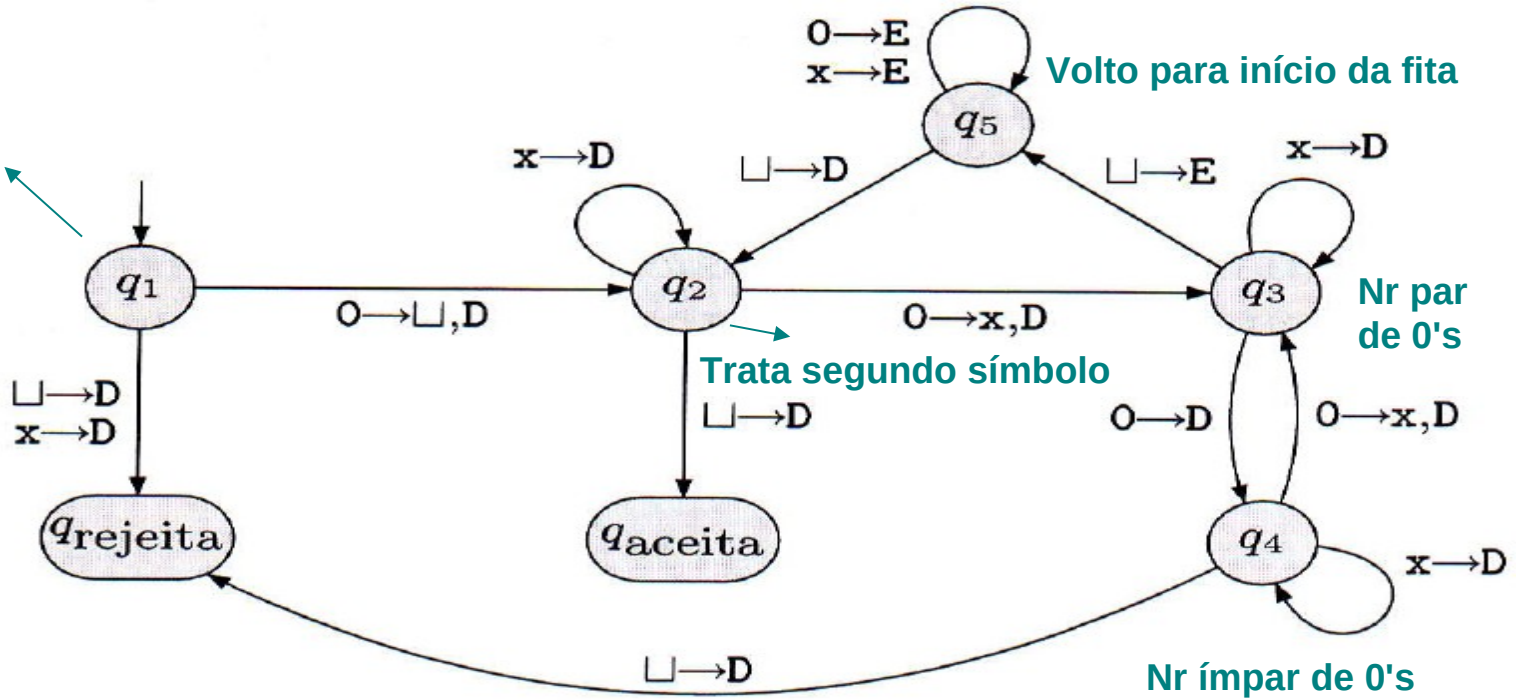
Exemplo para a cadeia 0000

$q_1 0000$
$\sqcup q_2 000$
$\sqcup x q_3 00$
$\sqcup x 0 q_4 0$
$\sqcup x 0 x q_3 \sqcup$
$\sqcup x 0 q_5 x \sqcup$
$\sqcup x q_5 0 x \sqcup$

$\sqcup q_5 x 0 x \sqcup$
$q_5 \sqcup x 0 x \sqcup$
$\sqcup q_2 x 0 x \sqcup$
$\sqcup x q_2 0 x \sqcup$
$\sqcup x x q_3 x \sqcup$
$\sqcup x x x q_3 \sqcup$
$\sqcup x x q_5 x \sqcup$

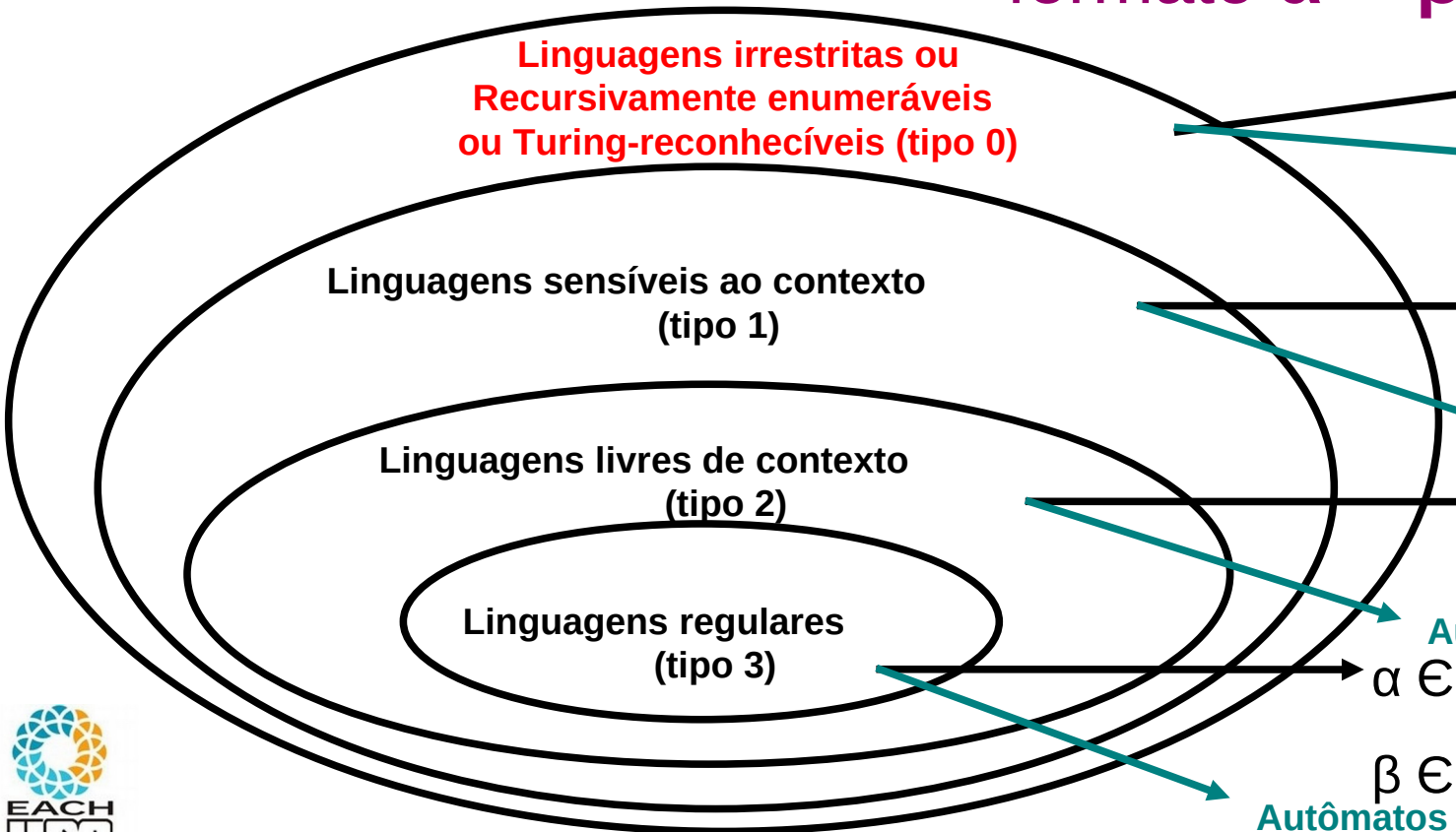
$\sqcup x q_5 x x \sqcup$
$\sqcup q_5 x x x \sqcup$
$q_5 \sqcup x x x \sqcup$
$\sqcup q_2 x x x \sqcup$
$\sqcup x q_2 x x \sqcup$
$\sqcup x x q_2 x \sqcup$
$\sqcup x x x q_2 \sqcup$
$\sqcup x x x \sqcup q_{aceita}$

Trata primeiro símbolo na primeira iteração (símbolo em branco colocado no lugar do primeiro 0 para indicar o início da fita)



Hierarquia de Chomsky

Gramáticas com produções no formato $\alpha \rightarrow \beta$



$\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$

$\beta \in (V \cup \Sigma)^*$

Máquina de Turing (fita ilimitada)

$\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$

$\beta \in (V \cup \Sigma)^*$

$|\alpha| <= |\beta|$

Máquina de Turing com fita limitada

$\alpha \in V$

$\beta \in (V \cup \Sigma)^*$

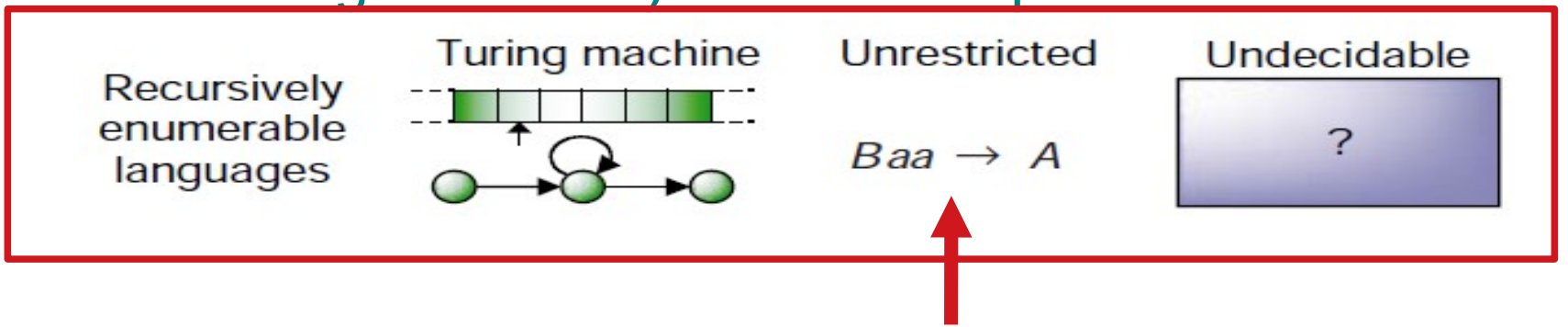
Autômatos com pilha

$\alpha \in V$

$\beta \in \Sigma_\varepsilon$, $\beta \in V$, $\beta \in (V\Sigma$ ou $\Sigma V)$

Autômatos finitos

Linguagens, modelos computacionais (dispositivos, gramáticas) e suas complexidades



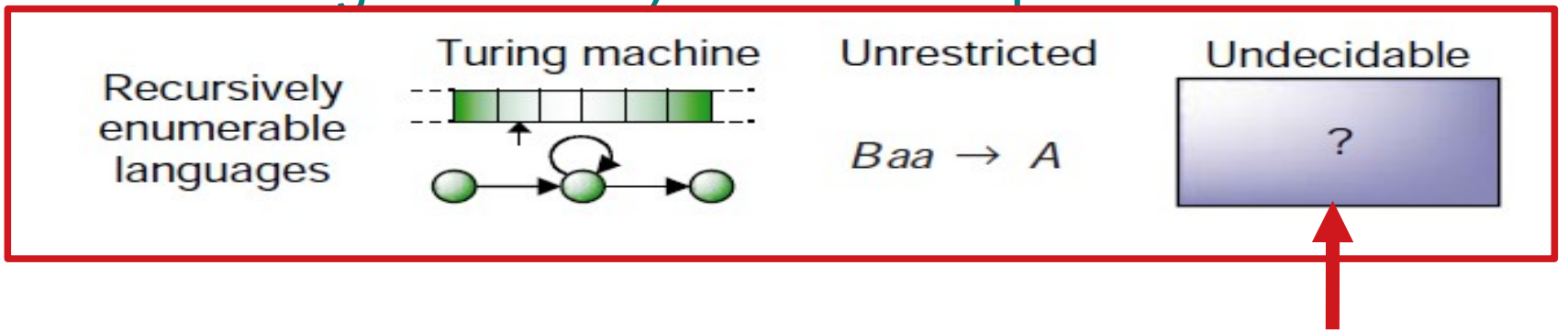
Máquinas de Turing são equivalentes a **gramáticas irrestritas**

Gramáticas irrestritas podem inclusive possuir o lado direito menor que o lado esquerdo

→ o tamanho das formas sentenciais geradas pode diminuir durante o processo de geração de uma cadeia

→ não há como garantir um número máximo de derivações

Linguagens, modelos computacionais (dispositivos, gramáticas) e suas complexidades

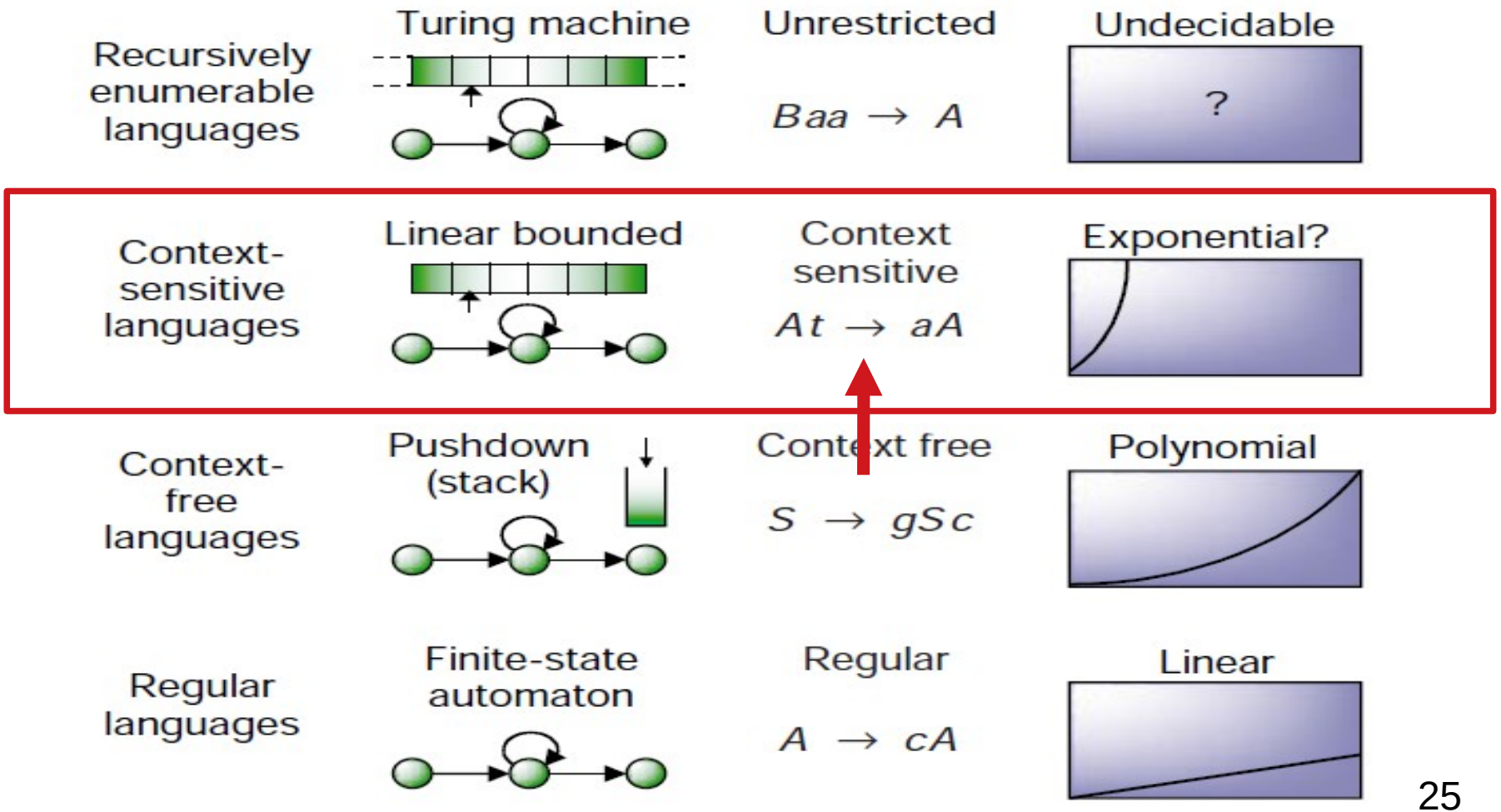


Falaremos mais disso em breve, mas tem a ver com o fato de que algumas MT podem entrar em loop, e não decidir a linguagem.

Logo, no caso geral, não podemos dizer que SEMPRE conseguimos DECIDIR uma uma linguagem recursivamente enumerável

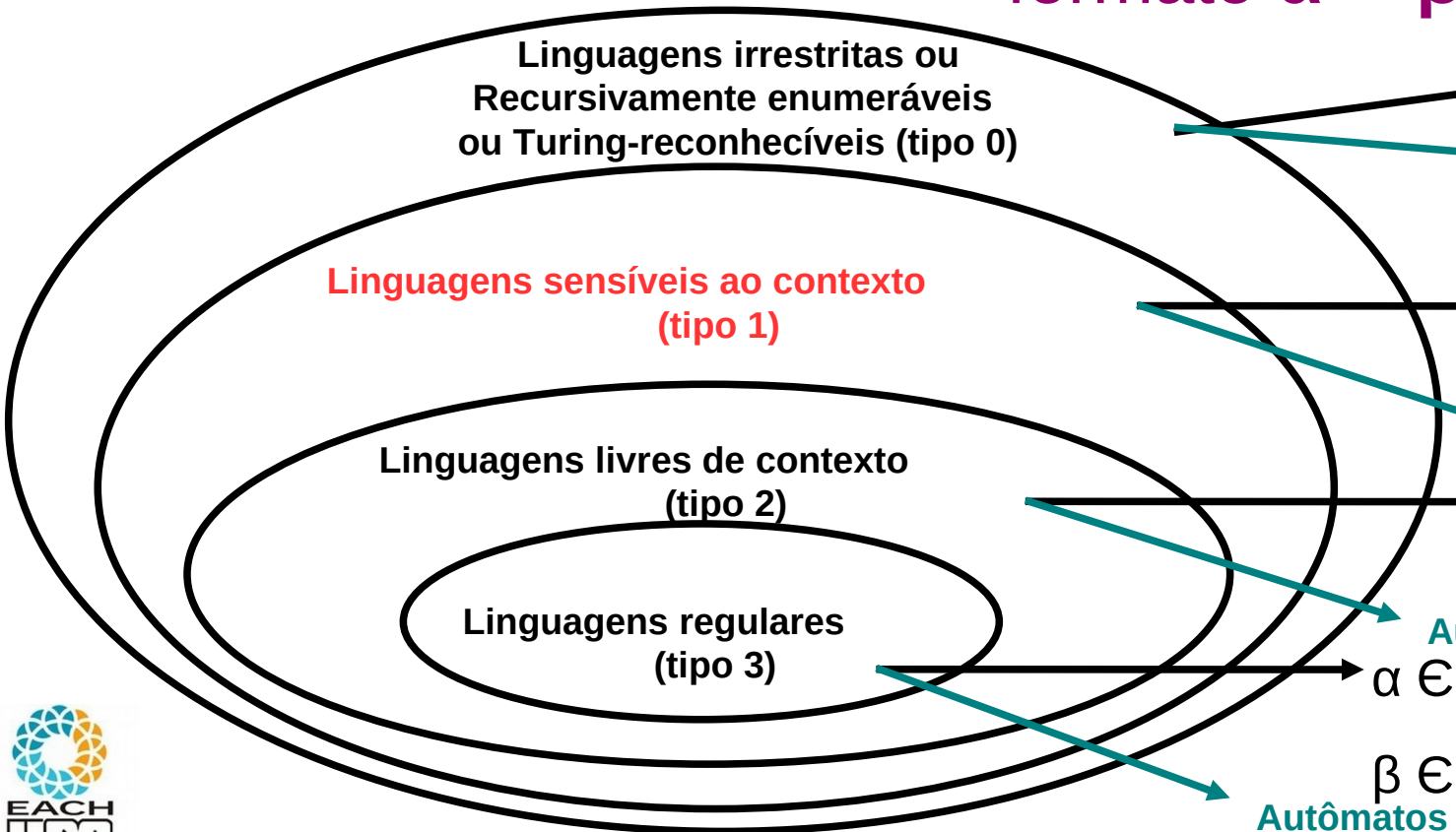
→ a classe de linguagens recursivamente enumeráveis é INDECIDÍVEL

Linguagens, modelos computacionais (dispositivos, gramáticas) e suas complexidades



Hierarquia de Chomsky

Gramáticas com produções no formato $\alpha \rightarrow \beta$



$\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$

$\beta \in (V \cup \Sigma)^*$

Máquina de Turing (fita ilimitada)

$\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$

$\beta \in (V \cup \Sigma)^*$

$|\alpha| <= |\beta|$

Máquina de Turing com fita limitada

$\alpha \in V$

$\beta \in (V \cup \Sigma)^*$

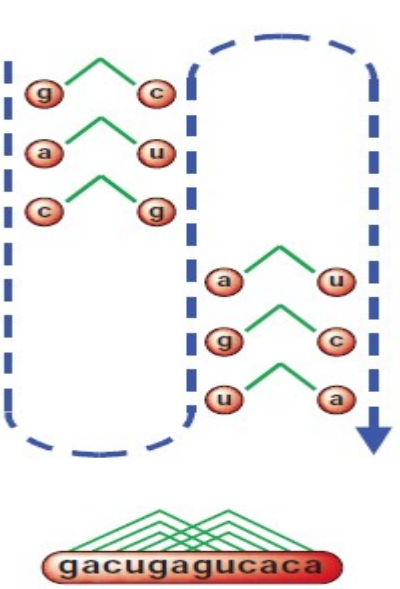
Autômatos com pilha

$\alpha \in V$

$\beta \in \Sigma_\varepsilon$, $\beta \in V$, $\beta \in (V\Sigma$ ou $\Sigma V)$

Autômatos finitos

Exemplo de problema sensível ao contexto: pseudonós em RNAs



Do artigo *The language of genes*, de David B. Searls (disponibilizado no edisciplinas)

Exemplo 1

Que linguagem essa gramática gera?

P = { S → aSBC

S → aBC,

CB → BC,

aB → ab,

bB → bb,

bC → bc,

cC → cc}

Exemplo 1

Que linguagem essa gramática gera?

P = { S → aSBC

S → aBC,

CB → BC,

aB → ab,

bB → bb,

bC → bc,

cC → cc}

$$L = \{a^n b^n c^n \mid n >= 1\}$$

Exemplo 2

Escreva uma gramática para a linguagem

L = { w | w possui a mesma quantidade de a's, b'c e c's}

Exemplo 2

Escreva uma gramática para a linguagem

L = { w | w possui a mesma quantidade de a's, b'c e c's}

P = { S → ABCS
S → ABC,
AB → BA,
AC → CA,
BA → AB,
BC → CB,
CA → AC,
CB → BC,
A → a,
B → b,
C → c }

Gramáticas e Linguagens sensíveis ao contexto

- Gramáticas sensíveis ao contexto (GSC) são monotônicas: o comprimento das formas sentenciais durante a derivação de uma sentença nunca sofre redução
- Rigorosamente, uma linguagem L é sensível ao contexto se e somente se:
  - ε não pertence a L e L = L(G), e G é GSC, ou
  - ε pertence a L e L – {ε} pode ser gerada por uma GSC

Gramáticas e Linguagens sensíveis ao contexto

- Se ε pertence a L, aceita-se colocar a regra $S \rightarrow \varepsilon$ se S for o símbolo inicial e S não aparecer do lado direito de nenhuma regra

- Uma linguagem é estritamente sensível ao contexto se ela não for livre de contexto

Linguagens sensíveis ao contexto

- **Teorema**: as Gramáticas Sensíveis ao Contexto e as **Máquinas de Turing com Fita Limitada** representam exatamente a mesma classe de linguagens – as linguagens sensíveis ao contexto
  - Demonstração:

RAMOS, M. V. M.; NETO, J. J.; VEGA, I. S. Linguagens Formais. Ed. Bookman, 2009 (seção 5.5)

Máquinas de Turing com fita limitada

Definição semelhante à da Máquina de Turing

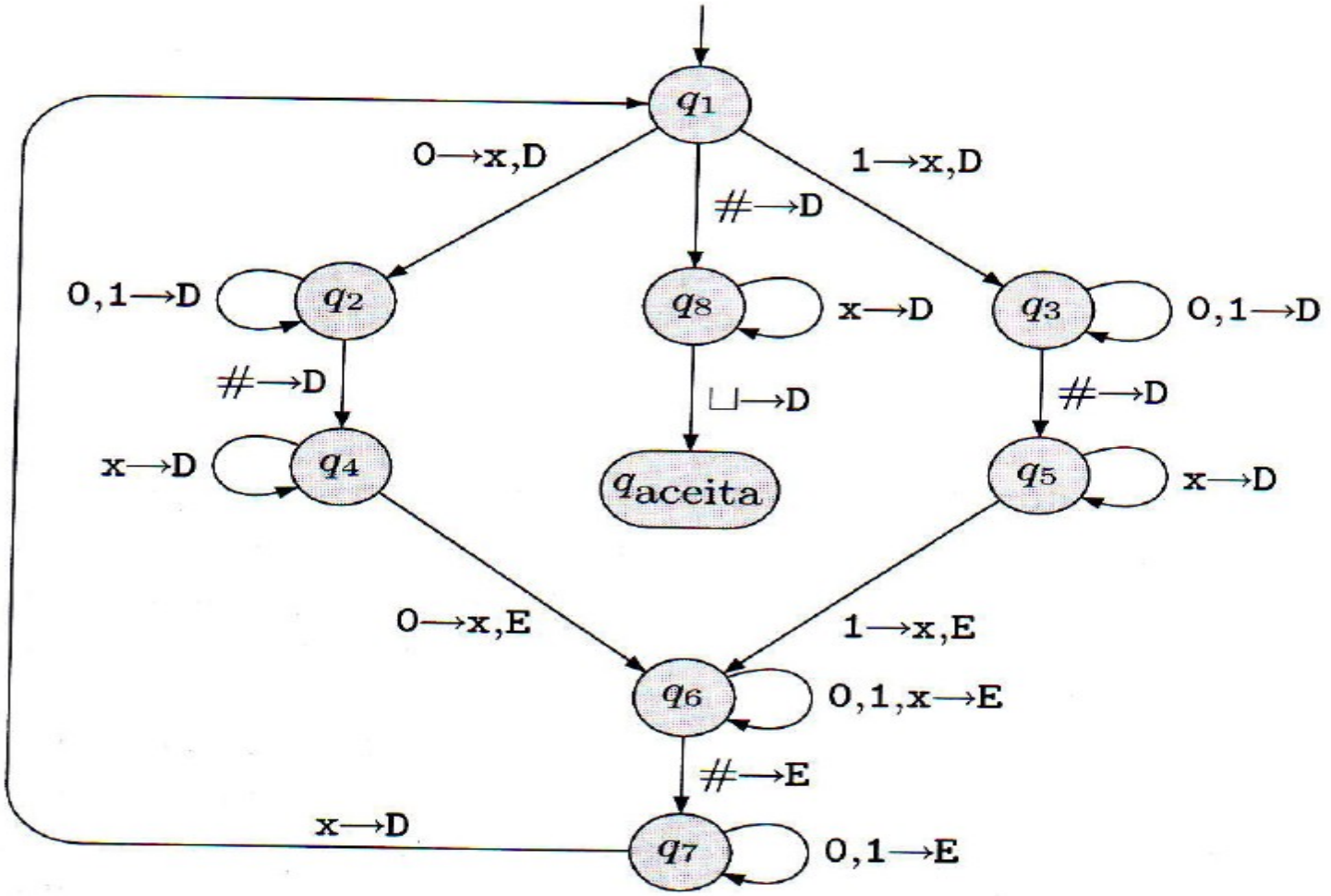
Diferença:

- Tamanho fita de trabalho = tamanho da entrada + 2
- Fita inicia e termina com símbolos delimitadores (por exemplo (“<” e “>”) não pertencentes ao alfabeto da linguagem

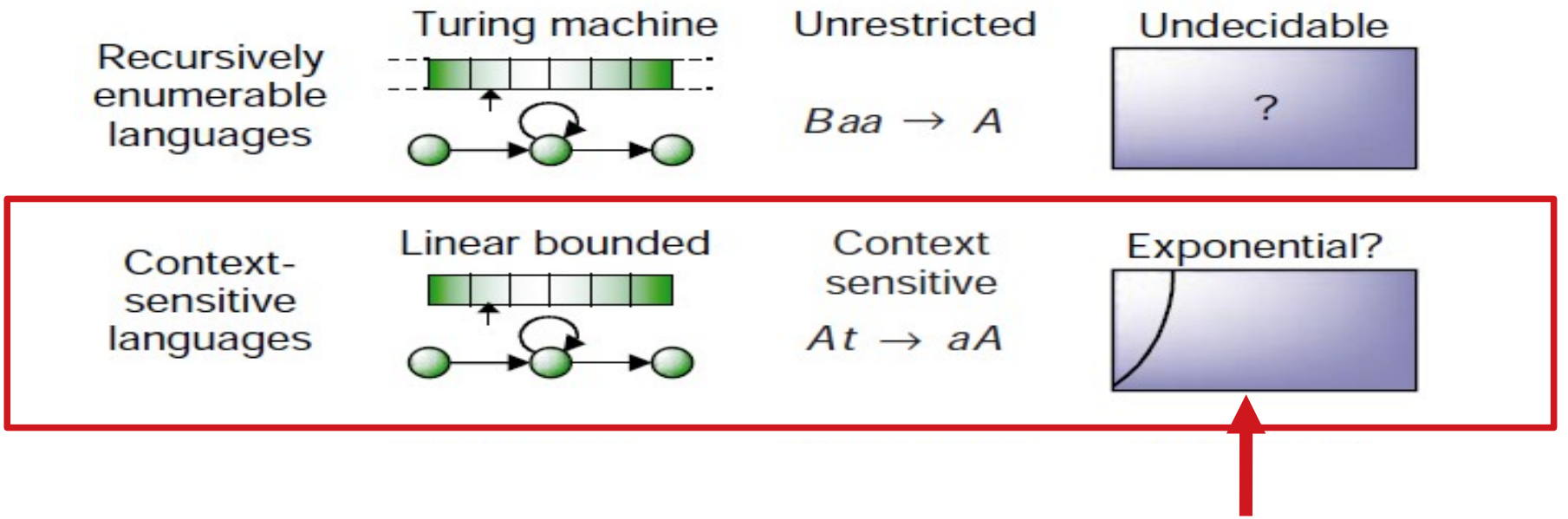
Exemplo

Adapte a Máquina de Turing que reconhece a linguagem B = {w#w | w pertence a {0,1}* }.

Máquina de Turing com fita ilimitada (o que teria que adaptar neste ex?)



Linguagens, modelos computacionais (dispositivos, gramáticas) e suas complexidades



Até o momento só se conhece algoritmos de reconhecimento (análise sintática) de gramáticas sensíveis ao contexto de tempo exponencial (no caso geral), mas não se provou que um algoritmo polinomial não seja possível…

Veremos no final do curso que este é um caso de problema NP-completo.

Exercícios

- Já podem fazer os exercícios 3.1 a 3.3, 3.5 e 3.8
- Pode ainda parecer estranho, mas vocês também já têm conhecimento suficiente para fazer os problemas 3.15 e 3.16. Dica: use a descrição de alto-nível de MTs.

Referências

RAMOS, M. V. M.; NETO, J. J.; VEJA, I. S. **Linguagens Formais: Teoria, Modelagem e Implementação**. Ed. Bookman, 2009. Cap 5.

SIPSER, M. Introdução à Teoria da Computação. Ed. Thomson. Cap 3.1