ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 18

Cap 3.2 – Variantes de MT (Enumeradores)

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

Aulas anteriores...

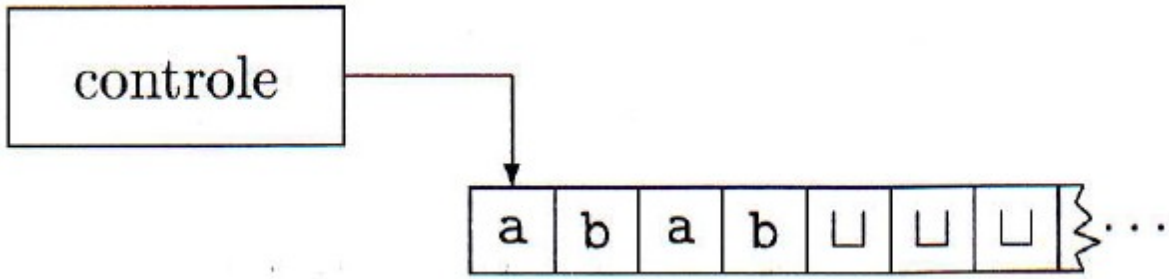
Máquinas de Turing – Definição formal

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

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

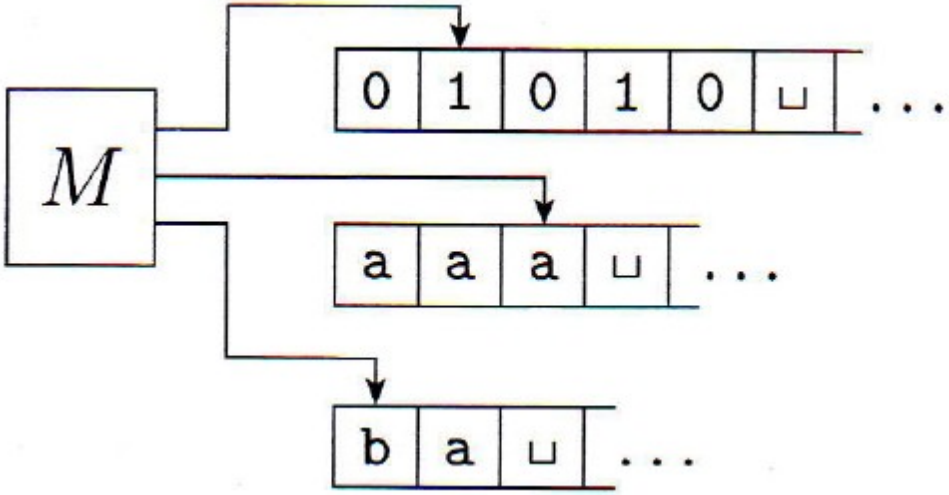
Máquinas de Turing



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)

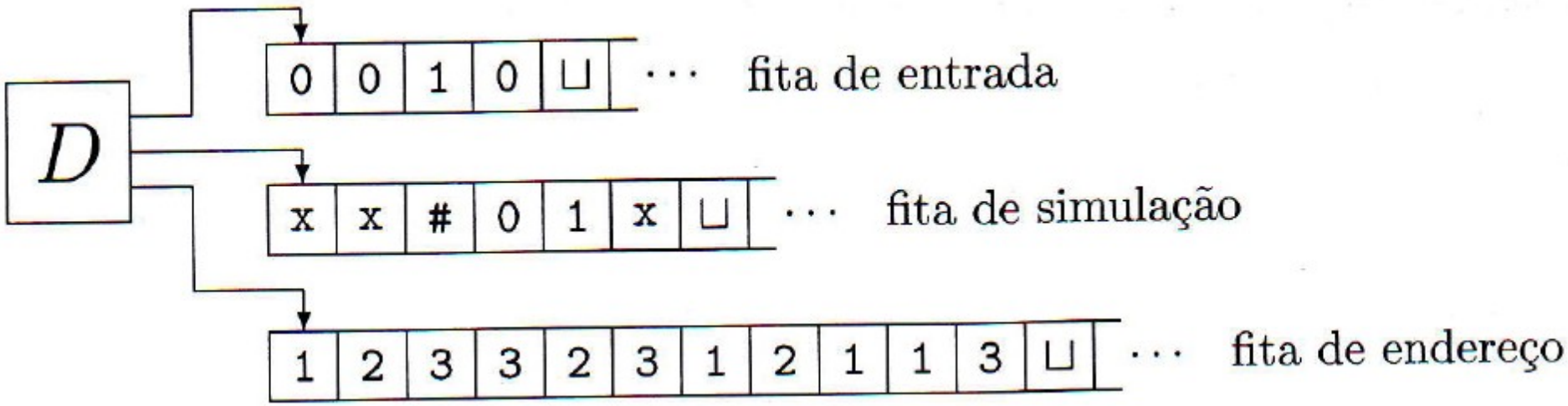
Máquinas de Turing Multifita



Máquinas de Turing Não-Determinísticas

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

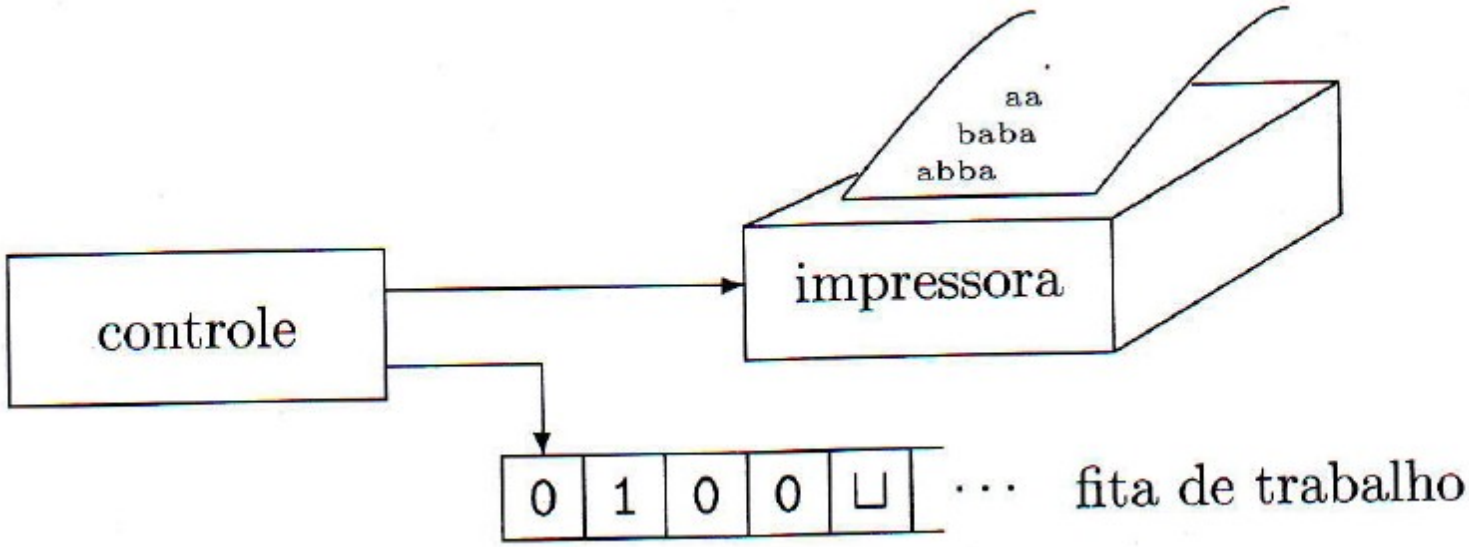
Simulação de uma MTND por uma MT Determinística:



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 de uma linguagem L

Uma MT M que utiliza E 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 de uma linguagem L

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

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 de uma linguagem L

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, w 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 de uma linguagem L

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, w 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, … 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 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 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, … 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

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

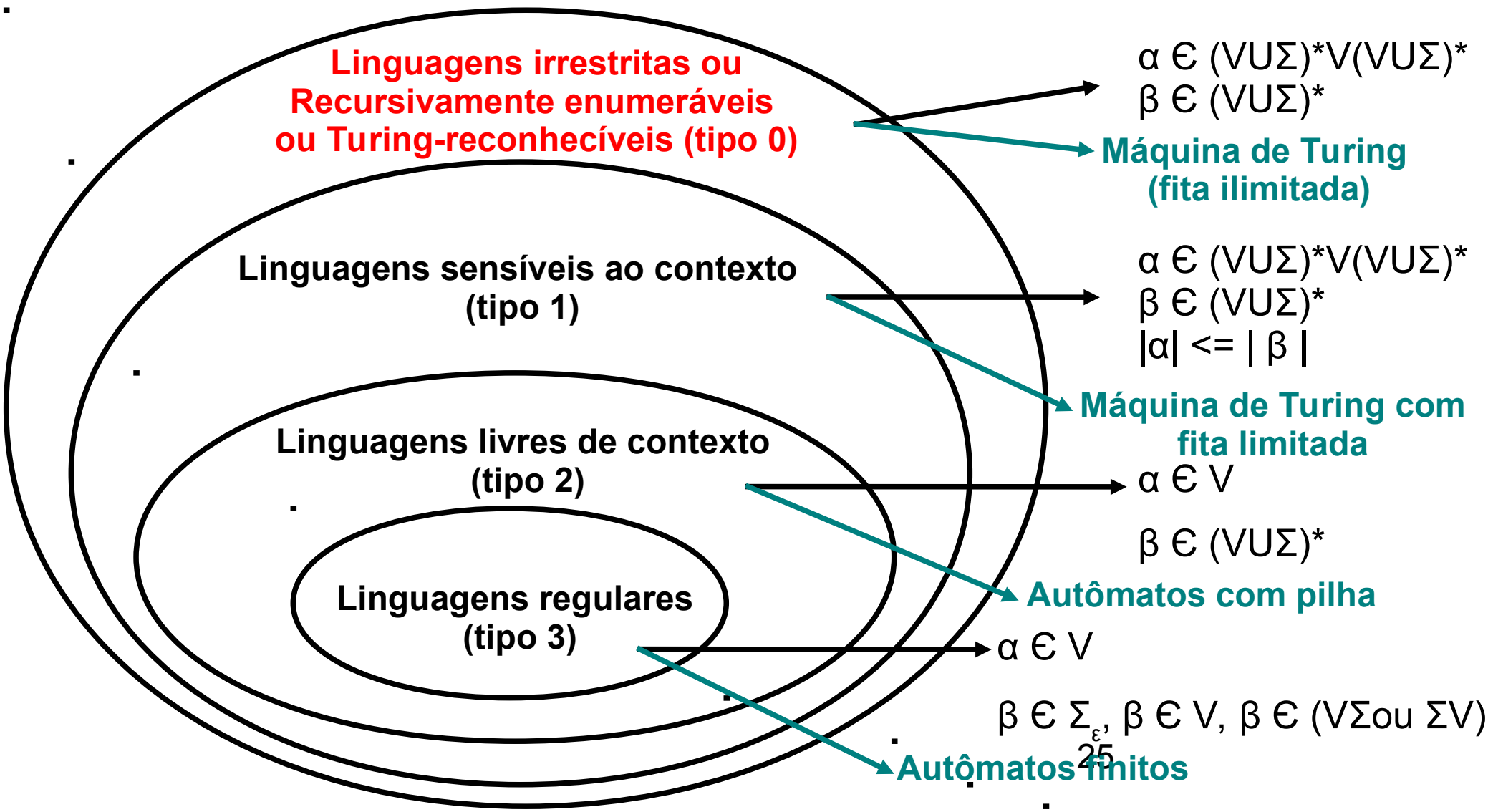
TEOREMA 3.21

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

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

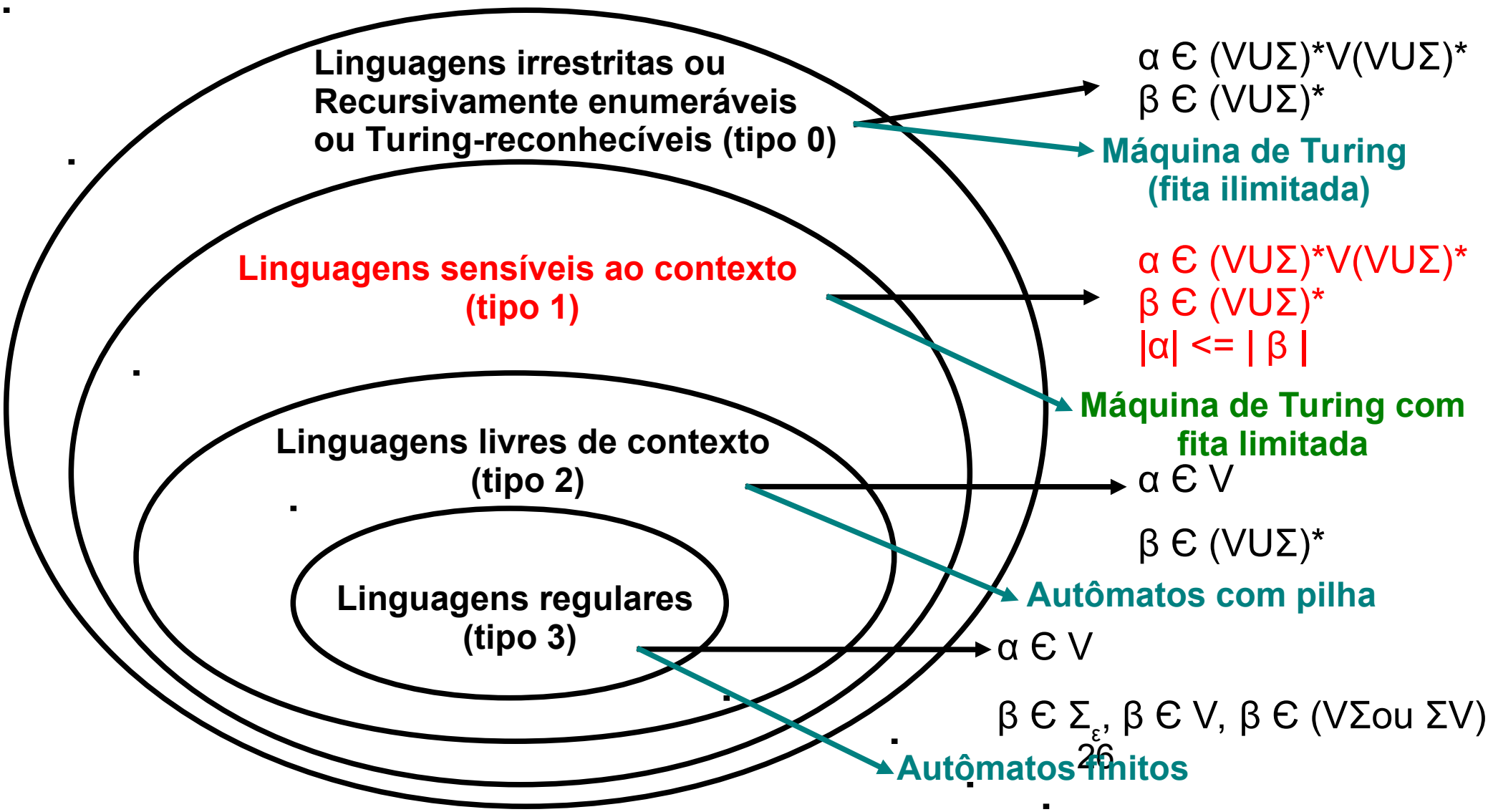
Hierarquia de Chomsky

Gramáticas com produções no formato **α →β**



Hierarquia de Chomsky

Gramáticas com produções no formato **α →β**



Exemplo 1

Que linguagem essa gramática gera?

$P = \{ S \rightarrow aSBC$

$S \rightarrow aBC,$

$CB \rightarrow BC,$

$aB \rightarrow ab,$

$bB \rightarrow bb,$

$bC \rightarrow bc,$

$cC \rightarrow cc\}$

Exemplo 1

Que linguagem essa gramática gera?

$P = \{ S \rightarrow aSBC$

$S \rightarrow aBC,$

$CB \rightarrow BC,$

$aB \rightarrow ab,$

$bB \rightarrow bb,$

$bC \rightarrow bc,$

$cC \rightarrow cc\}$

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

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), onde 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 ele não for livre de contexto

Máquinas de Turing com fita limitada

Definição semelhante à da Máquina de Turing

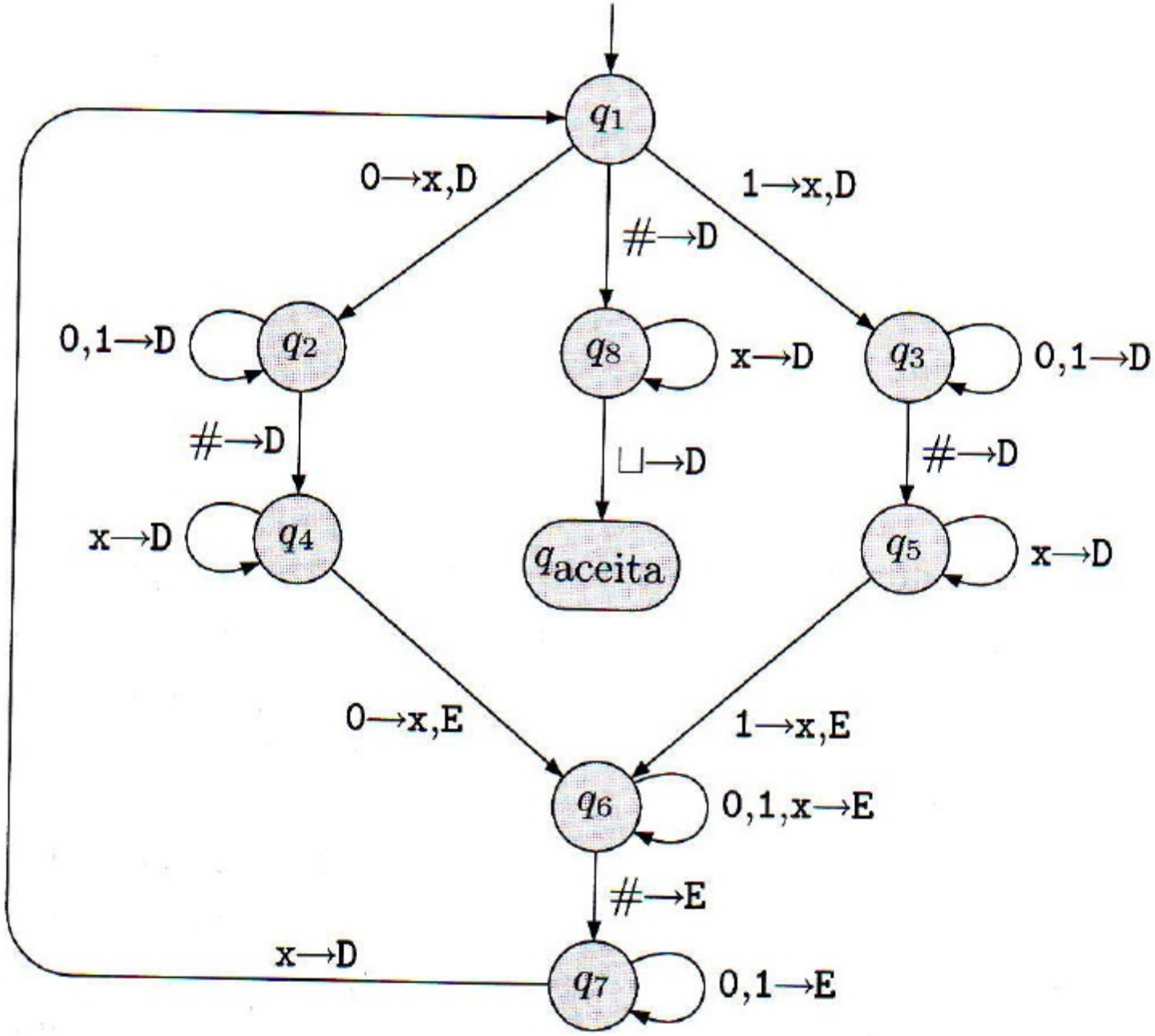
Diferença:

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

Exemplo

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

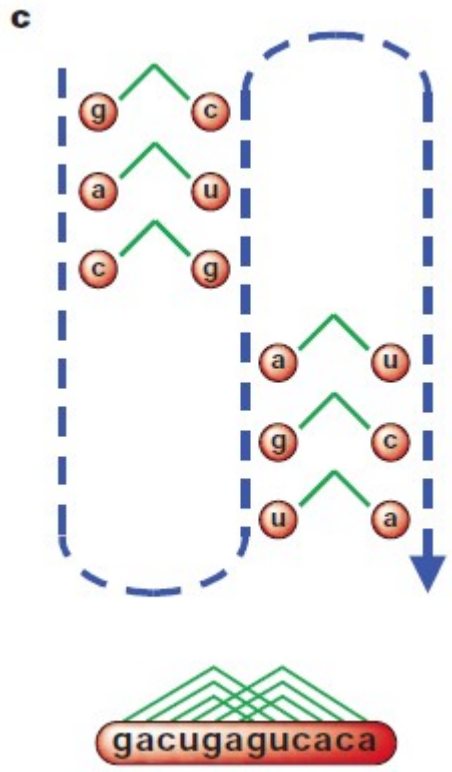
Máquina de Turing com fita ilimitada



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: livro (RAMOS, NETO e VEGA, 2009) – leitura complementar definida na primeira aula.

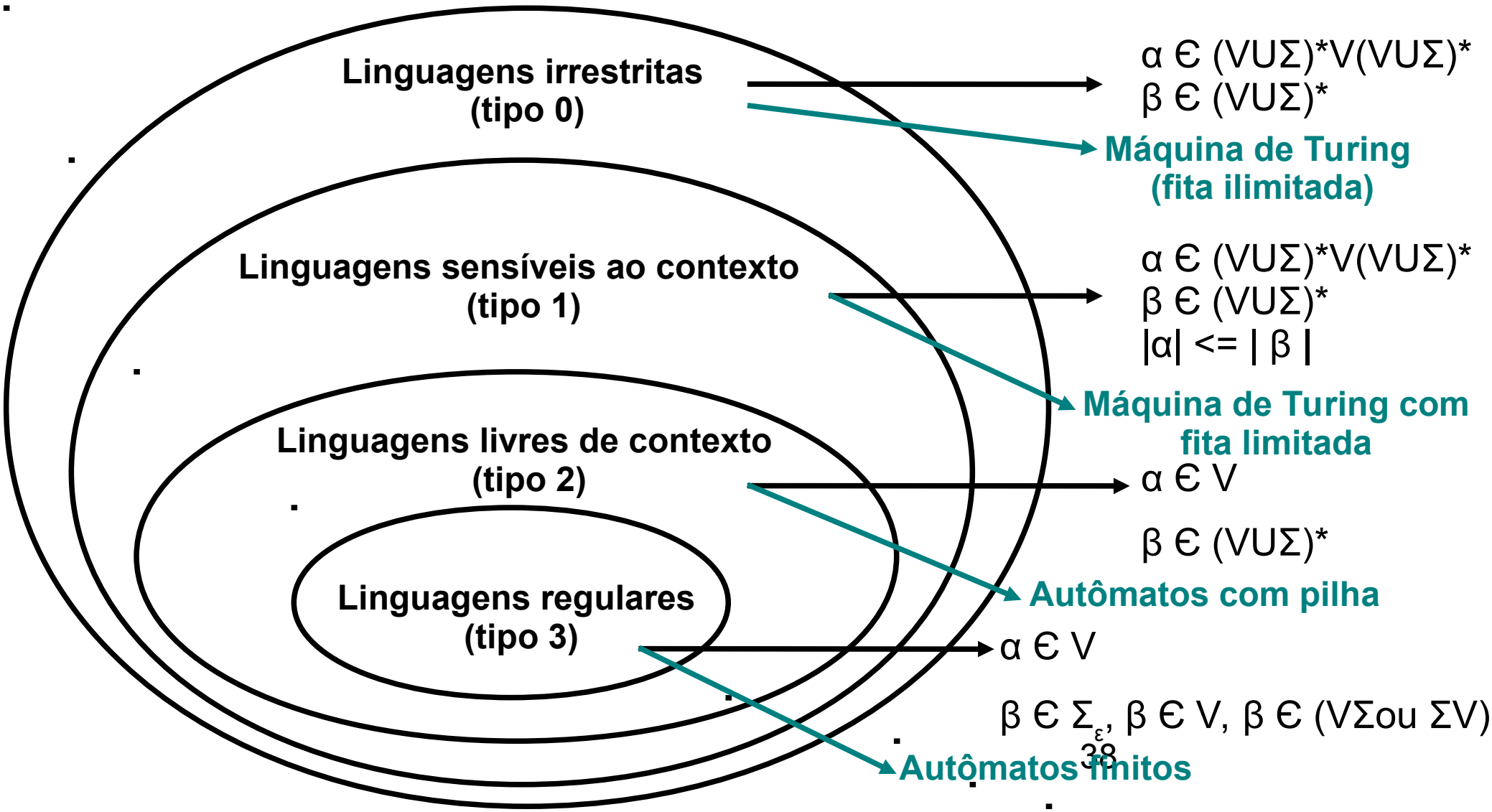
Exemplo de problema sensível ao contexto: pseudonós em RNAs



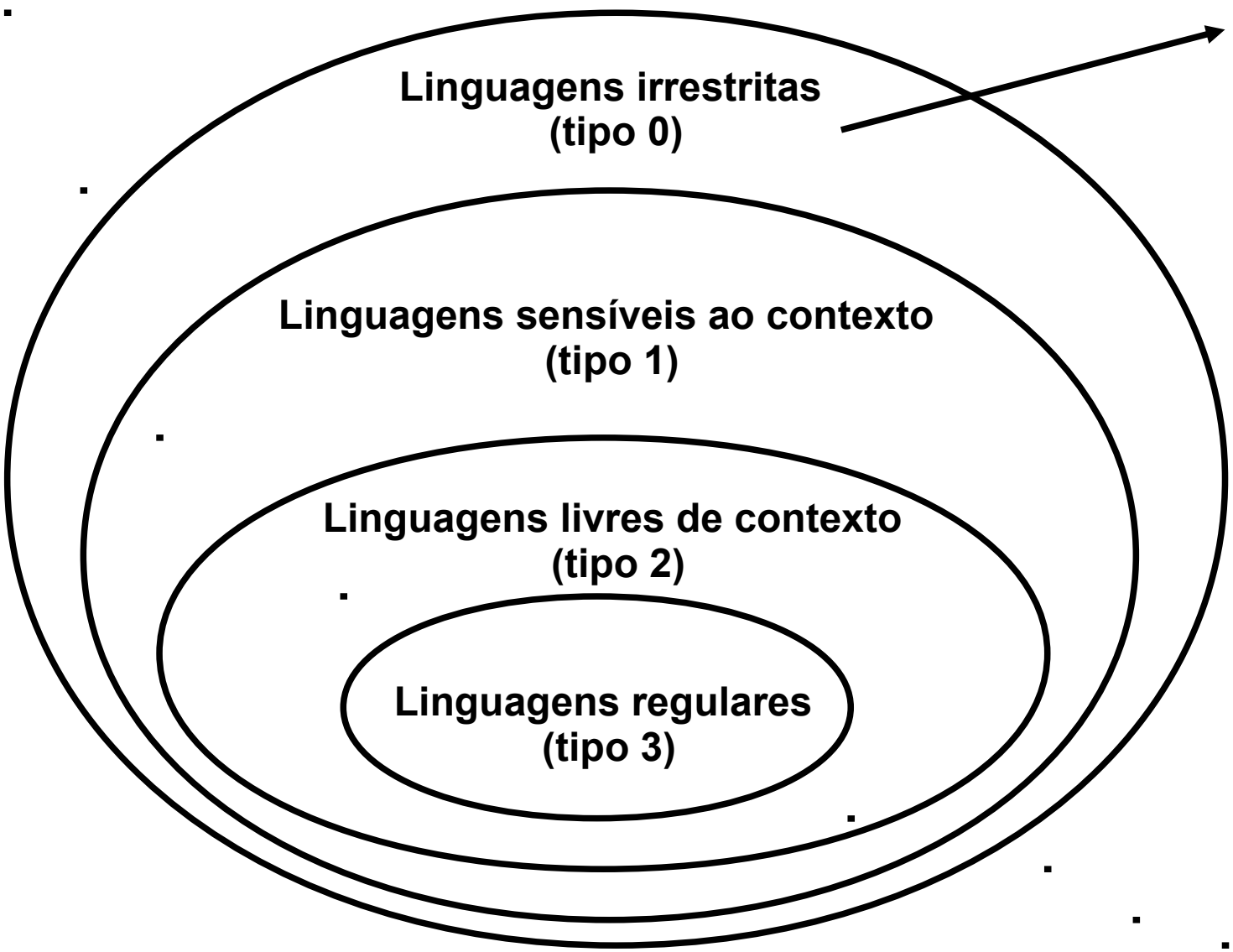
Do artigo *The language of genes*, de David B. Searls (disponibilizado no edisciplinas)

Hierarquia de Chomsky

$\alpha \rightarrow \beta$

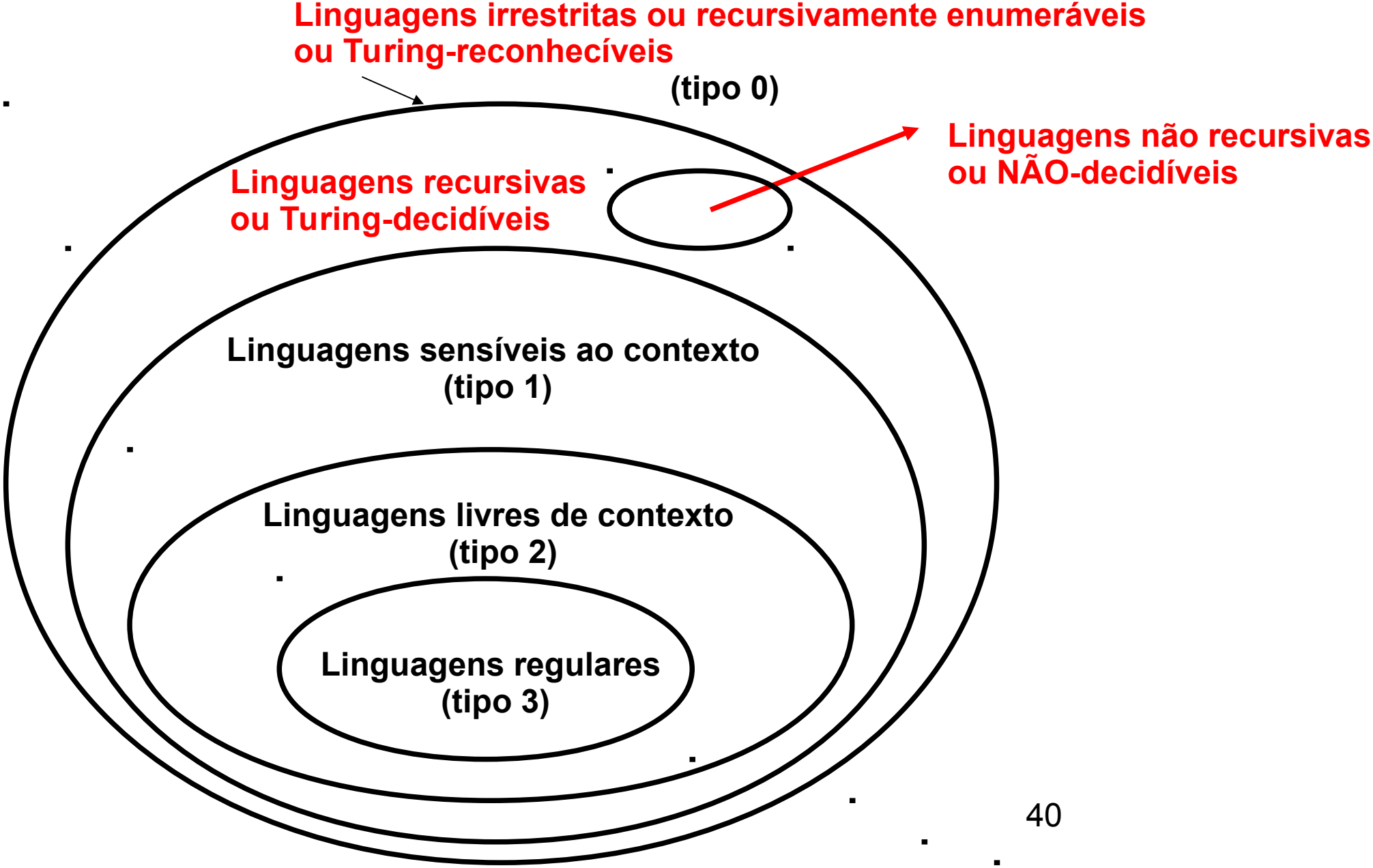


Hierarquia de Chomsky



Linguagens recursivas (Turing-decidíveis) e não recursivas (as demais)

Hierarquia de Chomsky



Linguagens irrestritas ou recursivamente enumeráveis ou Turing-reconhecíveis

- Uma linguagem L é chamada **irrestrita** ou **recursivamente enumerável** ou **Turing-reconhecível** se ela for **aceita** por pelo menos uma Máquina de Turing M, ou seja:
  - Para toda cadeia w Є L, M pára e aceita w
  - Para toda cadeia z Є Σ*-L, M pára e rejeita z ou executa uma sequência infinita de movimentações

Linguagens recursivas ou Turing-decidíveis

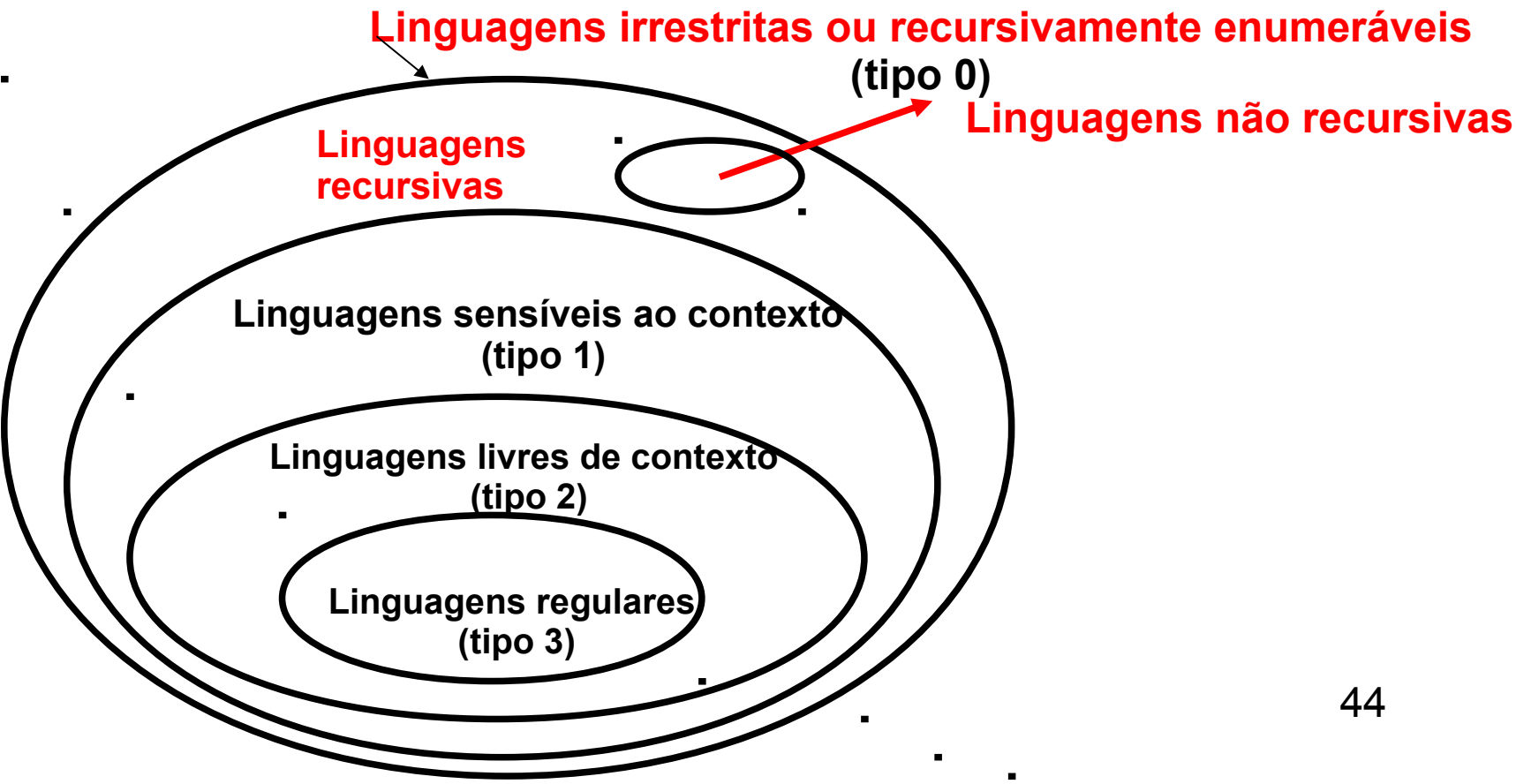
- Uma linguagem L é chamada recursiva se ela for **decidida** por pelo menos uma Máquina de Turing M, ou seja:
  - Para toda cadeia w Є L, M pára e aceita w
  - Para toda cadeia z Є Σ*-L, M pára e rejeita z

Linguagens não-recursivas

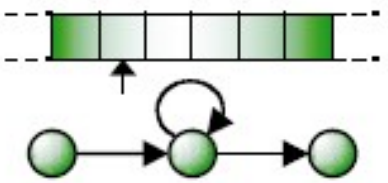
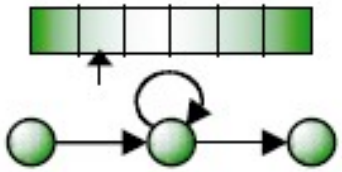
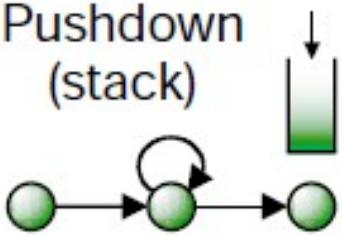
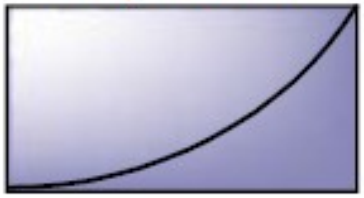
- Uma linguagem L é chamada não-recursiva se ela for **aceita** por pelo menos uma Máquina de Turing M **mas não decidida** por nenhuma Máquina de Turing, ou seja:
  - Para toda cadeia w Є L, M pára e aceita w
  - Para pelo menos uma cadeia z Є Σ*-L, M executa uma sequência infinita de movimentações (e pára e rejeita as demais cadeias y Є Σ*-L)

Linguagens recursivas

- Toda linguagem sensível ao contexto é recursiva
- Toda linguagem recursiva que NÃO é sensível ao contexto é também chamada estritamente recursiva



Linguagens, dispositivos, gramáticas e complexidades

Recursively enumerable languages	Turing machine 	Unrestricted $Baa \rightarrow A$	Undecidable
Context-sensitive languages	Linear bounded 	Context sensitive $At \rightarrow aA$	Exponential?
Context-free languages	Pushdown (stack) 	Context free $S \rightarrow gSc$	Polynomial 
Regular languages	Finite-state automaton	Regular $A \rightarrow cA$	Linear

Lista MÍNIMA cap 3

- 1a, 1d, 2d, 2e, 6, 8, 15, 16