

# Teoria da Computação

Prof. Diego Buchinger  
diego.buchinger@outlook.com  
diego.buchinger@udesc.br

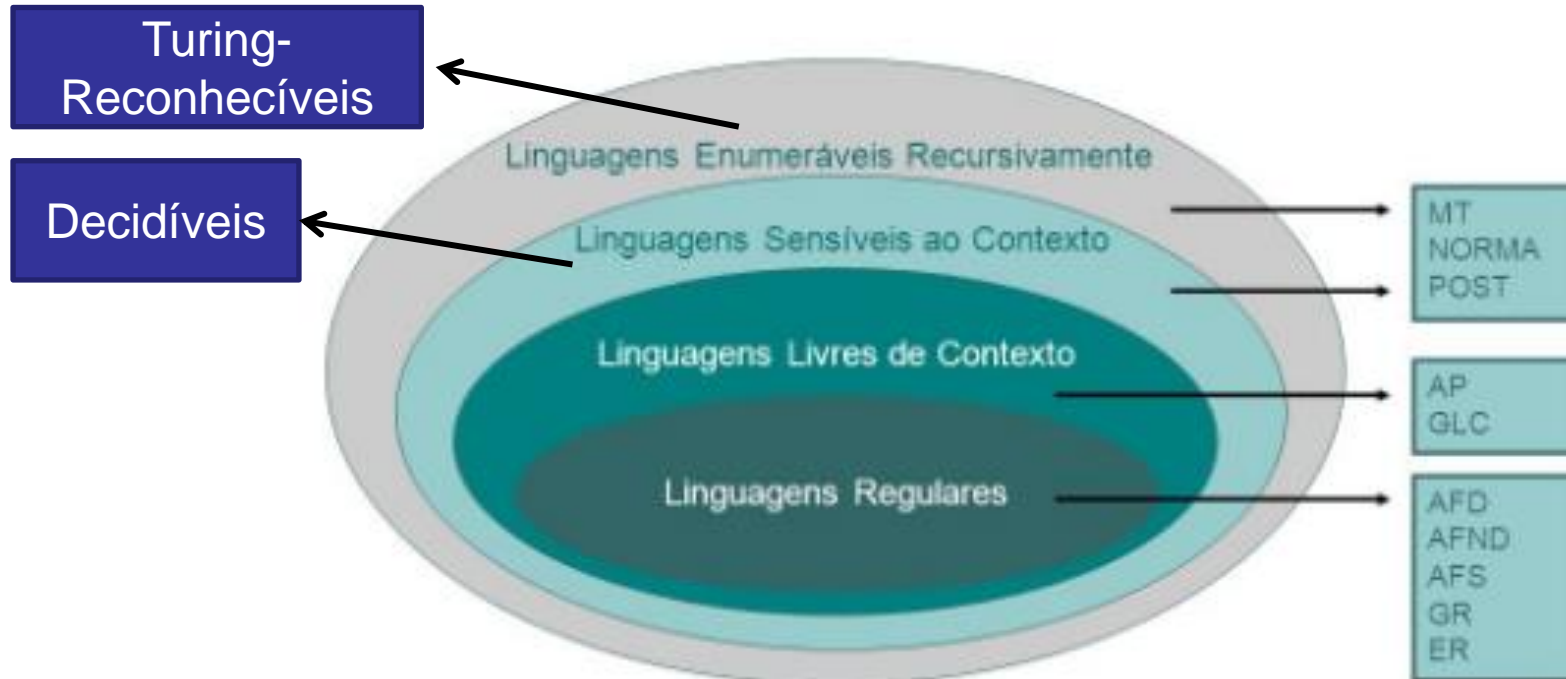
SIPSER, Michael.  
Introdução à Teoria da Computação

---

# Revisitando a Hierarquia de Chomsky

---

# Hierarquia de Chomsky



Existe alguma linguagem que não seja  
Recursivamente Enumerável,  
isto é, esteja fora deste conjunto?

# Linguagens Turing-reconhecíveis

---

**Corolário:** Algumas linguagens não são Turing-reconhecíveis. Elas são Turing-irreconhecíveis.

Prova:

1. Mostrar que o conjunto de todas as MTs é contável
2. Mostrar que o conjunto de todas as linguagens é incontável

# Linguagens Turing-reconhecíveis

---

**P1:** Mostrar que o conjunto de todas as MTs é contável

- $\Sigma^*$  o conjunto de todas as cadeias de tamanho finito sobre o alfabeto  $\Sigma$  e é contável:

{cadeia de comprimento 0, cadeias de comprimento 1,  
cadeias de comprimento 2, ... , cadeias de comprimento  $n$ }

**Lembrete:** um algoritmo ou MT possuem um número finito de passos.

Como uma MT  $M$  pode ser representada por uma codificação usando  $\Sigma^*$ , então a quantidade de MTs também é contável.

(ainda desconsiderando codificações inválidas de MTs – ex: MT sempre começam com 0)

# Linguagens Turing-reconhecíveis

---

**P2:** Mostrar que o conjunto de todas as linguagens é incontável

- Considere o conjunto de todas as sequências binárias infinitas, que chamaremos de  $B$ ;
- Provamos que  $B$  é incontável utilizando o método da diagonalização; [prove!]
- Considere também  $\mathcal{L}$  como o conjunto de todas as linguagens sobre o alfabeto  $\Sigma$ .
- Mostramos que  $\mathcal{L}$  é incontável apresentando uma correspondência com  $B$  (=os dois são do mesmo tamanho)
  - Mostrar que cada linguagem  $A \in \mathcal{L}$  tem uma sequência única em  $B$ . Criamos essa correspondência com  $f: \mathcal{L} \rightarrow B$

# Linguagens Turing-reconhecíveis

---

Assim, como o número de linguagens é incontável e o número de MT é contável, podemos afirmar que existem linguagens sem uma MT correspondente.

Em outras palavras, algumas linguagens não são Turing Reconhecíveis.

# Linguagens Decidíveis

---

**Teorema:** uma linguagem é decidível se, e somente se, ela é Turing-Reconhecível e co-Turing-Reconhecível.

O que é uma linguagem co-Turing-Reconhecível?

**Prova:** mostrar a equivalência nos dois sentidos

1. Mostrar que  $A_{\text{decidível}} \rightarrow A \text{ e } \bar{A} \text{ são Turing Reconhecíveis}$
2. Mostrar que  $A \text{ e } \bar{A} \text{ são Turing Reconhecíveis} \rightarrow A_{\text{decidível}}$



# Linguagens Decidíveis

---

**P1.** Mostrar que  $A_{\text{decidível}} \rightarrow A$  e  $\bar{A}$  são *Turing Reconhecíveis*  
Fácil percepção!

**P2.** Mostrar que  $A$  e  $\bar{A}$  são *Turing Reconhecíveis*  $\rightarrow A_{\text{decidível}}$   
Criar um MT que combina  $M_1$  – um reconhecedor para  $A$   
e  $M_2$  – um reconhecedor para  $\bar{A}$ .

Mas  $M_1$  ou  $M_2$  por serem Turing Reconhecíveis  
(e não Decidíveis) não podem entrar em loop?  
Como  $M$  vai ser decidível neste caso?

**Corolário:**  $\overline{A_{MT}}$  não é Turing-Reconhecível.

Sabemos que  $A_{MT}$  é Turing-Reconhecível

Considerando a prova anterior, se também fosse Turing-Reconhecível, o que poderíamos afirmar sobre  $A_{MT}$ ?

# Redutibilidade

---

# Redução ( $\leq$ )

---

**Redução:** é uma maneira de converter um problema em outro de forma que uma solução para o segundo problema possa ser usada para resolver o primeiro.

Exemplos:

se orientar na cidade C  $\leq$  conseguir o mapa da cidade

viajar para Las Vegas  $\leq$  comprar passagem aérea

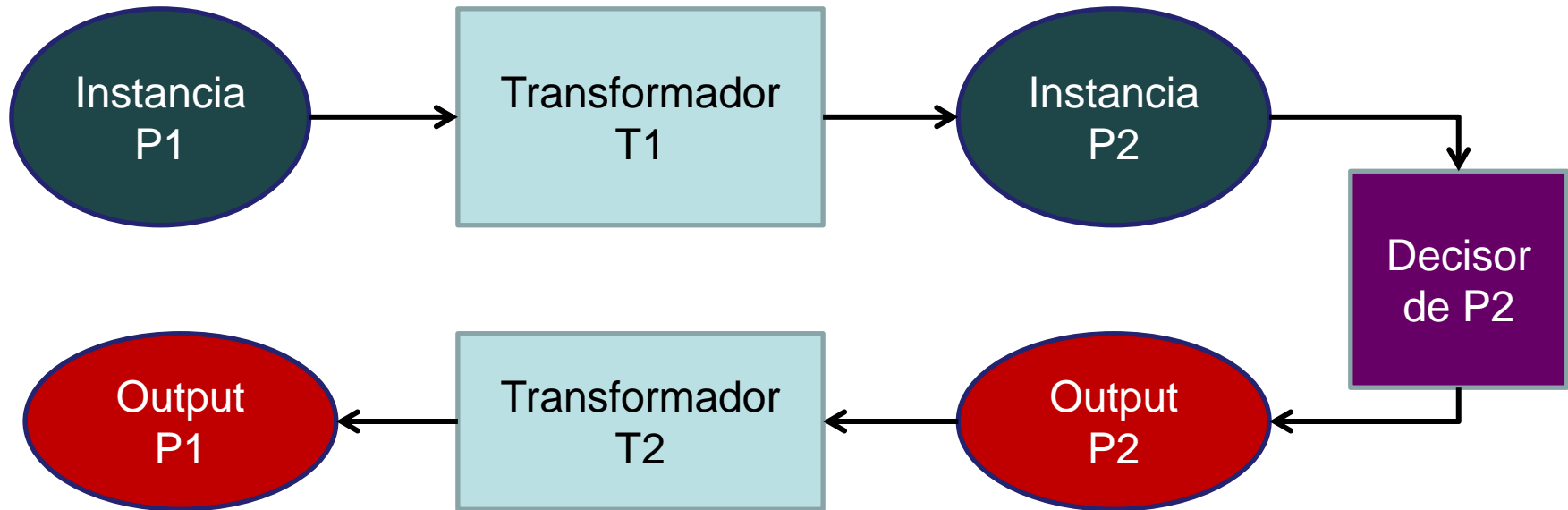
comprar passagem aérea  $\leq$  ganhar dinheiro

ganhar dinheiro  $\leq$  encontrar um trabalho

resolver um sistema linear  $\leq$  inverter uma matriz (mat. inversa)

# Redução ( $\leq$ )

---



Sendo A reduzível a B ( $A \leq B$ ), o que podemos afirmar sobre a dificuldade sobre A (em relação a B)?

## Algumas implicações lógicas:

Sendo  $A \leq B$  e  $B$  decidível  
o que podemos afirmar sobre  $A$ ?

Sendo  $A \leq B$  e  $A$  indecidível  
o que podemos afirmar sobre  $B$ ?

# Problemas Indecidíveis

---

Já provamos que  $A_{MT}$  é indecidível. Agora, podemos utilizá-lo para mostrar a indecidibilidade de outros problemas.

**Teorema:**  $PARA_{MT}$  é indecidível

$A_{MT} \Rightarrow$  Aceitador de MTs

$PARA_{MT} = \{\langle M, w \rangle \mid M \text{ é uma MT e } M \text{ pára sobre a entrada } w\}$

$PARA_{MT}$  é o real problema da parada

Prova por contradição: supomos que  $PARA_{MT}$  seja decidível (existe uma MT  $R$  que a decide) e usamos essa suposição para mostrar que  $A_{MT}$  é decidível (contradição com a prova de que  $A_{MT}$  é indecidível)

# Problemas Indecidíveis

---

**Teorema:**  $V_{MT}$  é indecidível

$$V_{MT} = \{\langle M \rangle \mid M \text{ é uma MT e } L(M) = \emptyset\}$$

$V_{MT}$  = problema da  
vacuidade de MT

Prova por contradição: supomos que  $V_{MT}$  é decidível (existe uma MT que decide se outra máquina de Turing  $M$  nada reconhece) e usamos essa suposição para mostrar que  $A_{MT}$  é decidível (contradição com a prova de que  $A_{MT}$  é indecidível)



# Problemas Indecidíveis

---

**Teorema:**  $\text{REGULAR}_{\text{MT}}$  é indecidível

$\text{REGULAR}_{\text{MT}} = \{\langle M \rangle \mid M \text{ é uma MT e } L(M) \text{ é uma linguagem regular}\}$

$\text{REGULAR}_{\text{MT}}$  = determinar se uma dada MT reconhece uma linguagem que pode ser reconhecida também por um modelo computacional mais simples.

Prova por contradição: supomos que  $\text{REGULAR}_{\text{MT}}$  é decidível (existe uma MT que decide se outra máquina de Turing  $M$  reconhece uma linguagem regular) e usamos essa suposição para mostrar que  $A_{\text{MT}}$  é decidível (contradição com a prova de que  $A_{\text{MT}}$  é indecidível)

# Problemas Indecidíveis

---

**Teorema:**  $EQ_{MT}$  é indecidível

$$EQ_{MT} = \{\langle M_1, M_2 \rangle \mid M_1 \text{ e } M_2 \text{ são MTs e } L(M_1) = L(M_2)\}$$

$EQ_{MT}$  = determinar se duas MTs são equivalentes, ou seja, se elas reconhecem a mesma linguagem.

Prova por contradição: supomos que  $EQ_{MT}$  é decidível (existe uma MT que decide se duas outras MTs  $M_1$  e  $M_2$  reconhecem a mesma linguagem) e usamos essa suposição para mostrar que  $V_{MT}$  é decidível (contradição com a prova de que  $V_{MT}$  é indecidível)

# Histórias de Computação

**Definição:** a história de computação para uma máquina de Turing  $M$  sobre uma entrada é a sequência de configurações  $(C_1, C_2, \dots, C_n)$  pelas quais a máquina passa à medida que processa uma entrada  $w$ , onde  $C_1$  é a configuração inicial e  $C_n$  é uma configuração de aceitação ou rejeição.

## Constituição de uma Configuração:

- Estado do controle
- Posição do cabeçote
- Conteúdo da fita

Representação de configurações:

$q_0bab \Rightarrow \#q_1ab \Rightarrow \#aq_2b \Rightarrow \dots$

Escreva uma MT det. para decidir a linguagem  
 $L_1 = \{w | w \in \{a, b\}^* \text{ e } w = xax \text{ com } x \in \{b\}^*\}$

Escreva uma história de computação considerando:

$w_1 = a$

$w_2 = bab$

$w_3 = bbab$

## Propriedades:

- Histórias de computação são finitas: se  $M$  não para sobre  $w$ , nenhuma história de computação de aceitação ou rejeição existe para este caso;
- MT deterministas possuem no máximo uma história de computação sobre uma dada entrada  $w$ ;
- MT não-deterministas podem ter muitas histórias de computação sobre uma determinada entrada – correspondente aos vários ramos de computação.

# Autômato Linearmente Limitado

---

Autômato Linearmente Limitado (ALL) é um modelo de MT limitada

## **Propriedades:**

- Cabeça de leitura-escrita não pode se mover para fora da parte da fita contendo a entrada (se a MT tentar fazer esse movimento inválido, a cabeça permanecerá onde está);
- O uso de um alfabeto de fita maior do que o alfabeto da entrada permite que a memória disponível seja incrementada de, no máximo, um fator constante, i.e. a quantidade de memória disponível é linear em  $n$ .

Pode-se entender o ALL como uma MT com quantidade limitada de memória. Logo, ela só pode resolver problemas que requerem memória que podem caber dentro da fita de entrada.

# Autômato Linearmente Limitado

---

Apesar de parecer limitado, ALLs são bastante poderosos!!

Toda LLC pode ser decidida por um ALL  
(Linguagem Livre do Contexto)

Agora considere  $A_{ALL}$  como problema de se determinar se um ALL aceita sua entrada ou não.

$$A_{ALL} = \{\langle M, w \rangle \mid M \text{ é um ALL que aceita a cadeia } w\}$$

**Este problema é decidível ou indecidível?**

# Autômato Linearmente Limitado

---

**Teorema:**  $A_{ALL}$  é decidível

Para provarmos este teorema devemos pensar na quantidade máxima possível de configurações!

Considerando um ALL com  $q$  estados,  $g$  símbolos no alfabeto de fita e uma entrada  $w$  de tamanho  $n$  (tamanho da fita ou tamanho de memória), quantas configurações distintas existem?

**Teorema:**  $A_{ALL}$  é decidível

**Lema:** seja  $M$  um ALL com  $q$  estados,  $g$  símbolos no alfabeto de fita e uma memória de tamanho  $n$ , existem exatamente:

$qng^n$  configurações distintas

**Prova:** o algoritmo que decide  $A_{ALL}$  é:

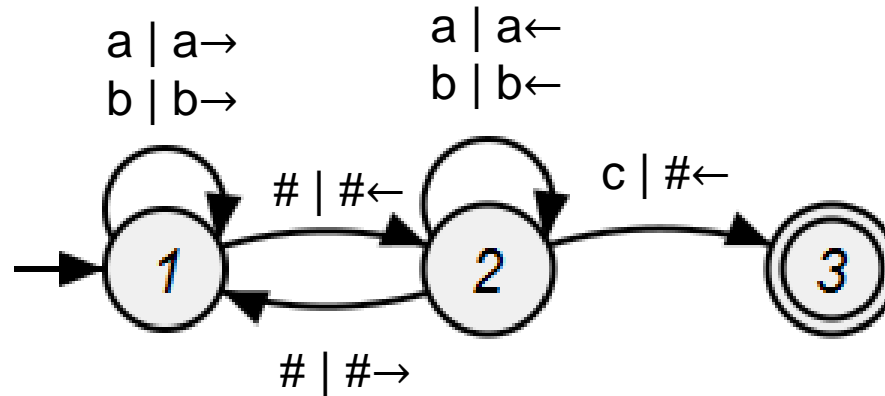
$L =$  “Sobre a entrada , onde  $M$  é um ALL e  $w$  é uma cadeia:

1. Simule  $M$  sobre  $w$  por  $qng^n$  passos ou até que ela pare.
2. Se  $M$  parou, *aceite* se ela aceitou e *rejeite* se ela rejeitou. Se ela não parou, *rejeite*.



# Autômato Linearmente Limitado

**Teorema:**  $A_{ALL}$  é decidível



$$\Sigma = \{a, b, c\}$$

# denota um marcador de início e final de fita

$w_1 = aba$  |03|

$w_2 = aababaaaabbaba$  |14|

# Problemas Indecidíveis

---

**Teorema:**  $V_{ALL}$  é indecidível

$$V_{ALL} = \{\langle M \rangle \mid M \text{ é um ALL onde } L(M) = \emptyset\}$$

$V_{ALL}$  = determinar se existe uma MT  $M$  que decide se, dado um ALL qualquer, este não aceita algo.

Prova por contradição: mostrar que se  $V_{ALL}$  fosse decidível então  $A_{MT}$  também seria (contradição com a prova de que  $A_{MT}$  é indecidível).

# Problemas Indecidíveis

---

**Teorema:**  $TODAS_{GLC}$  é indecidível

$$TODAS_{GLC} = \{\langle G \rangle \mid G \text{ é uma GLC e } L(G) = \Sigma^*\}$$

$TODAS_{GLC}$  = determinar se uma gramática-livre-do-contexto gera todas as cadeias possíveis.

Prova por contradição: supomos que  $TODAS_{GLC}$  seja decidível e então usamos esta suposição para mostrar que  $A_{MT}$  também seria decidível (contradição com a prova de que  $A_{MT}$  é indecidível).

# Um Problema Indecidível Simples

## Problema da Correspondência de Post

---