# TEC0001 – Teoria da Computação
# Videoaula 08
# Linguagens Indecidíveis

Karina Girardi Roggia

karina.roggia@udesc.br

Departamento de Ciência da Computação
Centro de Ciências Tecnológicas
Universidade do Estado de Santa Catarina

2020

# $A_{MT}$ é indecidível

$$A_{MT} = \{\langle M, w\rangle \mid M \text{ é uma máquina de Turing e } w \in L(M)\}$$

# Receita de bolo?

Quer se provar que $L$ é indecidível.

- Supõe-se que $L$ é **decidível**.
- Tem-se então um decisor $D$ para $L$.
- Utilizando $D$ como subrotina, é possível construir uma máquina de Turing $N$ que decide uma certa linguagem $H$ que já foi provada **indecidível**.
- Tal fato gera uma **contradição**!
- Conclui-se que a existência de um decisor $D$ para $L$ é impossível.

# Outras linguagens indecidíveis

- $PARA_{MT}$
- $V_{MT}$
- $REGULAR_{MT}$
- $EQ_{MT}$

# $PARA_{MT}$

$$PARA_{MT} = \{\langle M, w\rangle \mid M \text{ é máquina de Turing e para ao processar } w\}$$

**Prova:**

Supondo $D$ decisora de $PARA_{MT}$. Então pode-se construir a seguinte máquina de Turing.

$N =$ com entrada $\langle M, w\rangle$

1. Rode $D$ com entrada $\langle M, w\rangle$.
2. Se $D$ rejeitar, rejeite.
3. Se $D$ aceitar, rode $M$ com entrada $w$ e responda o que $M$ responder.

$N$ é um decisor de $A_{MT}$, o que é absurdo.

Logo, não existe decisor para $PARA_{MT}$.

$V_{MT} = \{\langle M \rangle \mid M \text{ é máquina de Turing e } L(M) = \varnothing\}$

**Prova:**

Supondo $D$ decisora de $V_{MT}$. Então pode-se construir um decisor $N$ para $A_{MT}$ usando $D$ como sub-rotina.

Para isso, $N$ construirá uma nova máquina de Turing a partir da entrada $\langle M, w\rangle$ dada. Tal máquina é como a seguir:

$X =$ com entrada $x$

1. Se $x \neq w$, rejeite.
2. Se $x = w$, rode $M$ com entrada $w$ e responda o que $M$ responder.

Temos então que

$$L(X) = \begin{cases} \{w\} & \text{se } w \in L(M) \\ \varnothing & \text{se } w \notin L(M) \end{cases}$$

O decisor de $A_{MT}$ poderia ser construído como segue:

$N =$ com entrada $\langle M, w \rangle$

1. Construa a máquina $X$ como definido anteriormente.
2. Rode $D$ com entrada $\langle X \rangle$.
3. Se $D$ aceitar, rejeite. Se $D$ rejeitar, aceite.

Não há como existir uma máquina que decida $A_{MT}$, portanto a suposição de que $D$ existe é falsa.

$REGULAR_{MT} =$
$\{\langle M\rangle \mid M$ é máquina de Turing e $L(M)$ é linguagem regular$\}$

**Prova:**
Supondo $D$ decisora de $REGULAR_{MT}$. Então pode-se construir um decisor $N$ para $A_{MT}$ usando $D$ como sub-rotina.
Para isso, $N$ construirá uma nova máquina de Turing a partir da entrada $\langle M, w\rangle$ dada. Tal máquina é como a seguir:

$X =$ com entrada $x$

1. Se $x$ está no formato $0^n1^n$, para $n \in \mathbb{N}$, aceite.
2. Caso contrário, rode $M$ com entrada $w$ e responda o que $M$ responder.

Temos então que

$$L(X) = \begin{cases} \{0^n1^n \mid n \in \mathbb{N}\} & \text{se } w \notin L(M) \\ \Sigma^* & \text{se } w \in L(M) \end{cases}$$

## $REGULAR_{MT}$

O decisor de $A_{MT}$ poderia ser construído como segue:

$N =$ com entrada $\langle M, w \rangle$

1. Construa a máquina $X$ como definido anteriormente.
2. Rode $D$ com entrada $\langle X \rangle$.
3. Se $D$ aceitar, aceite. Se $D$ rejeitar, rejeite.

Não há como existir uma máquina que decida $A_{MT}$, portanto a suposição de que $D$ existe é falsa.

# $EQ_{MT}$

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

**Prova:**
Seja a máquina de Turing $X$ a seguir, que aceita a linguagem vazia.


Então se existir um decisor $D$ para $EQ_{MT}$, a máquina a seguir decide $V_{MT}$.

$N =$ com entrada $\langle M\rangle$

1. Rode $D$ com entrada $\langle X, M\rangle$.
2. Se $D$ aceitar, aceite. Se $D$ rejeitar, rejeite.

Como não pode existir decisor para $V_{MT}$, a existência de $D$ é impossível.