DEFINIÇÃO 3.5

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

DEFINIÇÃO 3.6

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

TEOREMA 3.16

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

COROLÁRIO 3.18

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

COROLÁRIO 3.19

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

TEOREMA 3.21

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

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

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

$REGULAR_{\mathsf{MT}} = \{\langle M\rangle |\ M \text{ é uma MT e } L(M) \text{ é uma linguagem regular}\}.$

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

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

$TOD_{\mathsf{GLC}} = \{\langle G\rangle |\ G \text{ é uma GLC e } L(G) = \Sigma^*\}.$

$A_{\mathsf{MT}} = \{\langle M, w\rangle |\ M \text{ é uma MT e } M \text{ aceita } w\}.$

TEOREMA 4.22

Uma linguagem é decidível sse ela é Turing-reconhecível e co-Turing-reconhecível.

Em outras palavras, uma linguagem é decidível exatamente quando ela e seu complemento são ambas Turing-reconhecíveis.

========

COROLÁRIO 4.23

$\overline{A_{\mathsf{MT}}}$ não é Turing-reconhecível.

=====

DEFINIÇÃO 5.17

Uma função $f\colon \Sigma^* \longrightarrow \Sigma^*$ é uma ***função computável*** se alguma máquina de Turing M, sobre toda entrada w, pára com exatamente $f(w)$ sobre sua fita.

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

DEFINIÇÃO 5.20

A linguagem A é ***redutível por mapeamento*** à linguagem B, escrito $A \leq_{\mathrm{m}} B$, se existe uma função computável $f\colon \Sigma^* \longrightarrow \Sigma^*$, onde para toda w,

$$w \in A \iff f(w) \in B.$$

A função f é denominada a ***redução*** de A para B.

TEOREMA 5.22

Se $A \leq_{\mathrm{m}} B$ e B for decidível, então A é decidível.

COROLÁRIO 5.23

Se $A \leq_{\mathrm{m}} B$ e A for indecidível, então B é indecidível.

TEOREMA 5.28

Se $A \leq_{\mathrm{m}} B$ e B é Turing-reconhecível, então A é Turing-reconhecível.

COROLÁRIO 5.29

Se $A \leq_{\mathrm{m}} B$ e A não é Turing-reconhecível, então B não é Turing-reconhecível.

TEOREMA 5.30

EQ_{MT} não é nem Turing-reconhecível nem co-Turing-reconhecível.

DEFINIÇÃO 7.12

P é a classe de linguagens que são decidíveis em tempo polinomial sobre uma máquina de Turing determinística de uma-única-fita. Em outras palavras,

$$\mathrm{P} = \bigcup_k \mathrm{TIME}(n^k).$$

DEFINIÇÃO 7.19

NP é a classe de linguagens que têm verificadores de tempo polinomial.

TEOREMA 7.20

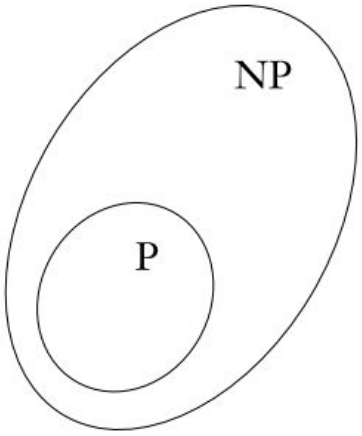
Uma linguagem está em NP sse ela é decidida por alguma máquina de Turing não-determinística de tempo polinomial.

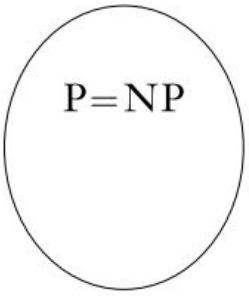
TEOREMA 7.24

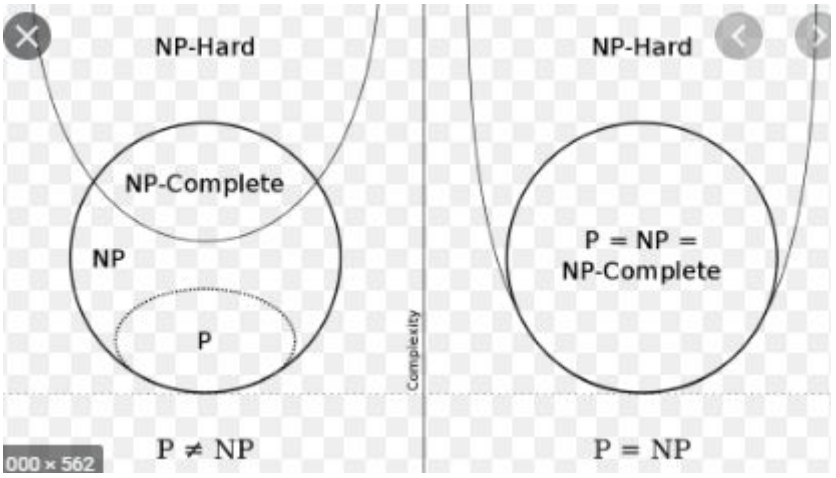
CLIQUE está em NP.

TEOREMA 7.25

SUBSET-SUM está em NP.







DEFINIÇÃO 7.28

Uma função $f: \Sigma^* \longrightarrow \Sigma^*$ é uma ***função computável em tempo polinomial*** se alguma máquina de Turing de tempo polinomial M existe que pára com exatamente $f(w)$ na sua fita, quando iniciada sobre qualquer entrada w.

DEFINIÇÃO 7.29

A linguagem A é ***redutível por mapeamento em tempo polinomial***,[1] ou simplesmente ***redutível em tempo polinomial***, à linguagem B, em símbolos $A \leq_P B$, se uma função computável em tempo polinomial $f: \Sigma^* \longrightarrow \Sigma^*$ existe, onde para toda w,

$$w \in A \iff f(w) \in B.$$

A função f é chamada ***redução de tempo polinomial*** de A para B.

TEOREMA 7.31

Se $A \leq_P B$ e $B \in \mathrm{P}$, então $A \in \mathrm{P}$.

DEFINIÇÃO 7.34

Uma linguagem B é ***NP-completa*** se ela satisfaz duas condições:

1. B está em NP, e
2. toda A em NP é redutível em tempo polinomial a B.

TEOREMA 7.35

Se B for NP-completa e $B \in \mathrm{P}$, então $\mathrm{P} = \mathrm{NP}$.

TEOREMA 7.36

Se B for NP-completa e $B \leq_P C$ para C in NP, então C é NP-completa.

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

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

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

AMT