

1. Pesquise outras definições sobre decidibilidade

- Khanacademy - Um **problema indecidível** é aquele que deve dar uma resposta "sim" ou "não", mas ainda não existe nenhum algoritmo que possa responder corretamente em todas as entradas.
- Geeks - Os problemas para os quais não podemos construir um algoritmo que possa responder ao problema corretamente em tempo finito são chamados de Problemas Indecidíveis.
- Wikipedia - Na teoria da computação e na teoria da complexidade computacional, um **problema indecidível** é um problema de decisão para o qual é provado ser impossível construir um algoritmo que sempre leva a uma resposta sim ou não correta

2. Como colocar esse problema na forma de linguagem? E como colocar alguma linguagem na forma de problema?

Problema: Dados dois números $x, y \in \mathbb{N}$, y é divisível por x ?

Linguagem: $L = \{w \in \mathbb{N} \text{ tal que } x, y \in \mathbb{N} \rightarrow y \geq x \text{ e } y \% x = 0\}$

Linguagem: $L1 = \{w \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^* \text{ tal que } |w| = 3\}$

Problema: Como podemos agrupar o conjunto das centenas?

3. Quantos problemas/linguagens existem? O conjunto de problemas/linguagens é enumerável?

(1) Existem infinitos problemas/linguagens;

- (2) Sendo Σ um alfabeto finito, **todos os problemas/linguagens serão subconjuntos de Σ^*** ;
- (3) Com base em (1) temos que o conjunto de todos os problemas/linguagens é o conjunto que une todos os subconjuntos de Σ^* , o $P(\Sigma^*)$;
- (4) Sabemos que Σ^* é enumerável, portanto, possui a cardinalidade dos números naturais;
- (5) Sabemos que $P(\Sigma^*)$ tem a cardinalidade maior que a dos números naturais e, portanto, é não enumerável;
- (6) Portanto, **o conjunto de todos os problemas/linguagens é não enumerável.**

4. O conjunto de programas em C (ou em qualquer outra linguagem de programação) é enumerável?

- (1) Cada programa de computador é uma sequência de palavras formadas por um alfabeto finito A;
- (2) Por (1), temos que o conjunto A é finito e, por conseguinte, enumerável;
- (3) Tendo em vista a afirmação (1), o programa de computador é um elemento da união $\bigcup_{i=1}^{\infty} A_i$ com $A_i = A \cup \{\varepsilon\}$, onde ε é a palavra vazia;
- (4) Dado que A é enumerável, A_i também é (como A_i contém apenas um elemento a mais comparado A);
- (5) A união de um número finito de conjuntos enumeráveis é enumerável e, portanto, $\bigcup_{i=1}^{\infty} A_i$ é enumerável;
- (6) Como todos os programas de computador são elementos de $\bigcup_{i=1}^{\infty} A_i$, o conjunto de todos os programas é um subconjunto do conjunto enumerável $\bigcup_{i=1}^{\infty} A_i$;
- (7) Como o subconjunto de um conjunto enumerável também é enumerável, segue que o **conjunto de todos os programas é enumerável.**

5. Por que problemas indecidíveis tem que existir?

Os problemas indecidíveis têm que existir, para que possamos vislumbrar avanços na computação, ou seja, vislumbrar até onde a computação pode chegar tendo em vista a classe de problemas indecidíveis que almejamos poderem ser dissolvidos por meio dela. De fato, os problemas indecidíveis promovem uma certa motivação. Esses problemas fazem com que os estudos sejam aprimorados cada vez mais.