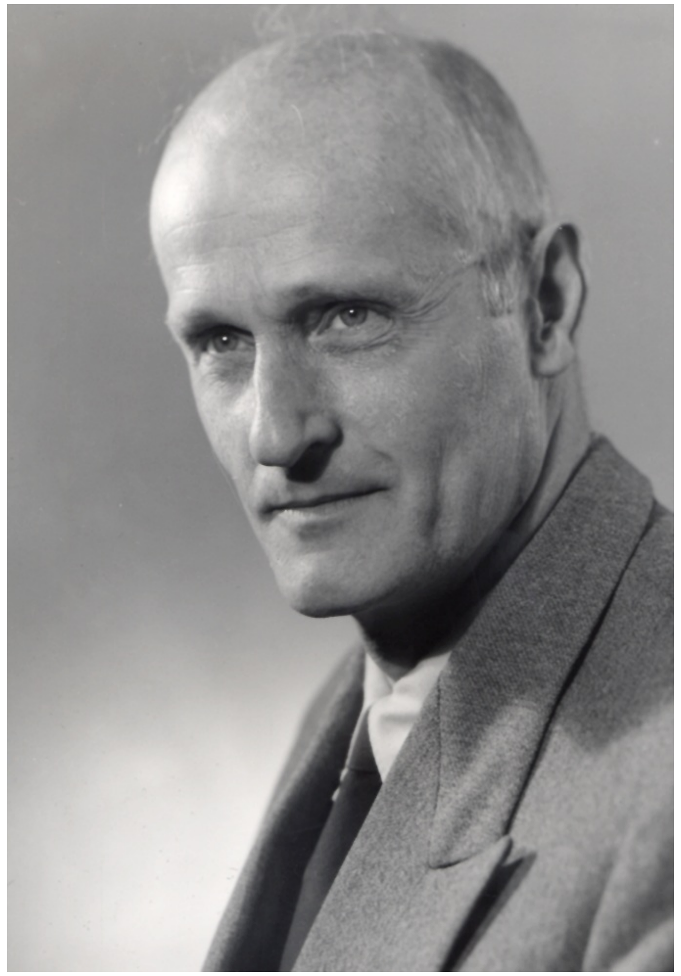
TEORIA DA COMPUTAÇÃO

0. Introdução

Alonzo Church



Stephen Kleene

Alan Turing

Prof. Abrantes Araújo Silva Filho

O que é Teoria da Computação? Quais seus objetivos?

Estudo das propriedades matemáticas fundamentais do hardware, do software e das aplicações de um computador.

Problema: aplicação prática difícil de enxergar e entender

Por ser uma disciplina essencialmente teórica, os alunos têm muita dificuldade de enxergar a aplicação imediata desse conhecimento. E com razão:

Gramáticas	→	projetar uma linguagem de programação
Autômatos Finitos	→	buscas em strings e casamento de padrões
Expressões regulares	→	buscas em strings e casamento de padrões
Problema difícil	→	complexidade, NP-completo
Provas matemáticas	→	?

Aplicação tardia:

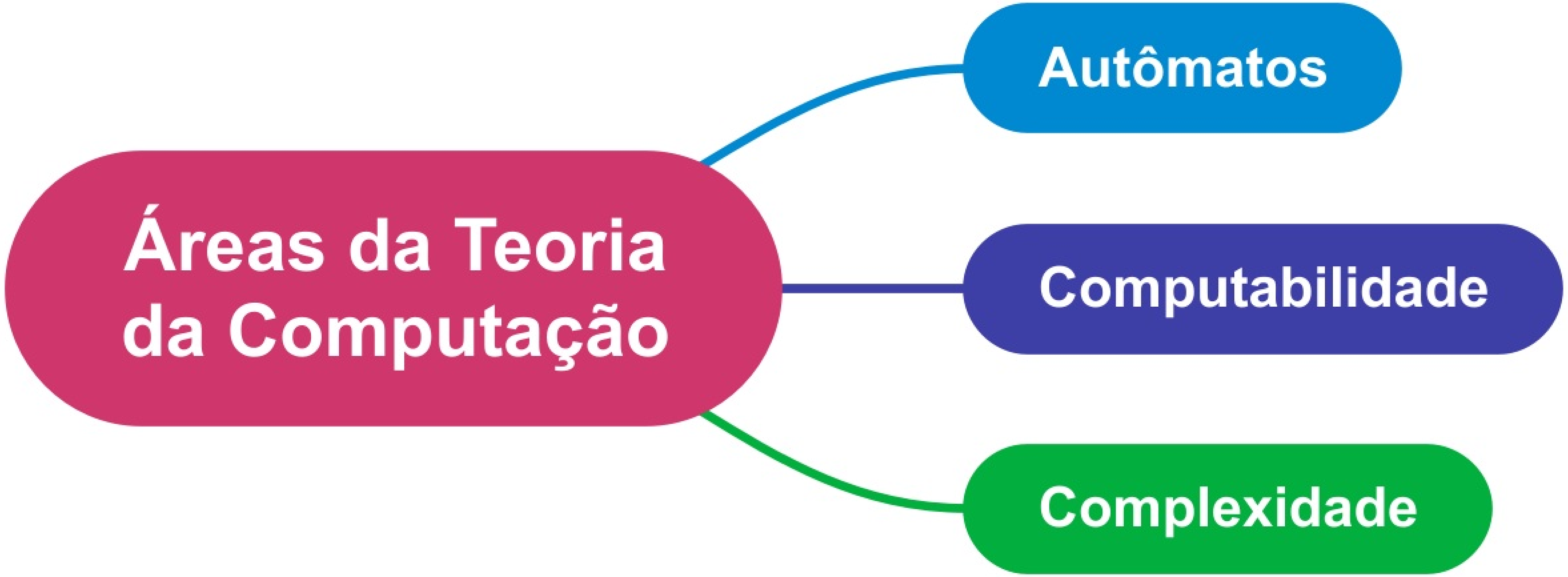
Aumenta capacidade mental, melhora capacidade de se expressar de modo claro e preciso, de resolver problemas e saber quando um problema é insolúvel.

Qual a melhor maneira de aprender? Qual nossa abordagem?

Fazer uma parte teórica pura com seminários semanais de aplicação prática em situações de programação e resolução de problemas.

Vai dar certo? Só Jesus sabe...

Áreas fundamentais da Teoria da Computação



Quais são as capacidades fundamentais e limitações dos computadores?

Teoria da Complexidade

O que faz com que alguns problemas sejam computacionalmente difíceis e outros sejam computacionalmente fáceis? Exemplos:

- → **Ordenação**
- → **Agendamento**

Resposta:

Teoria da Complexidade

O que faz com que alguns problemas sejam computacionalmente difíceis e outros sejam computacionalmente fáceis? Exemplos:

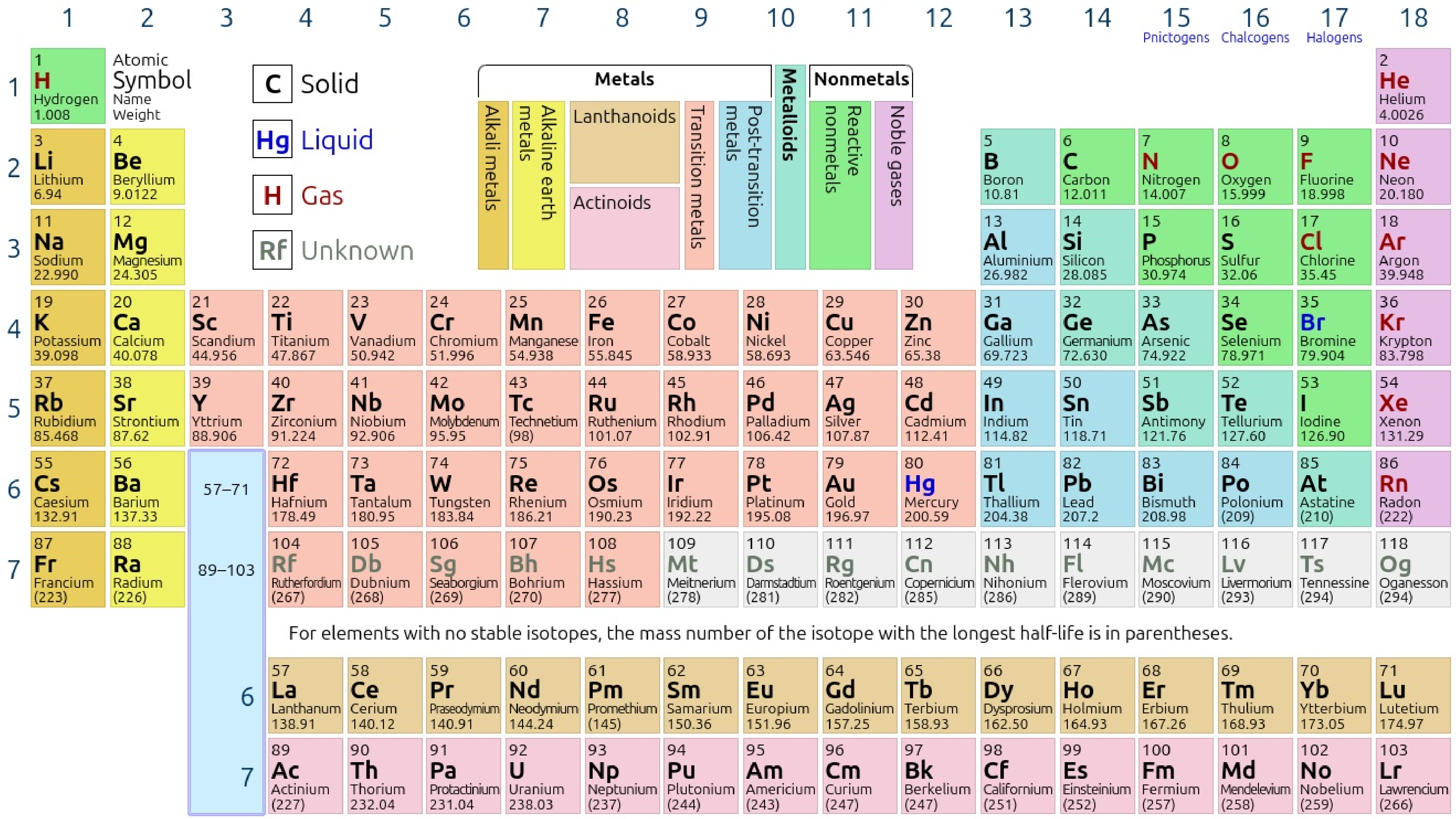
- → **Ordenação**
- → **Agendamento**

Resposta:

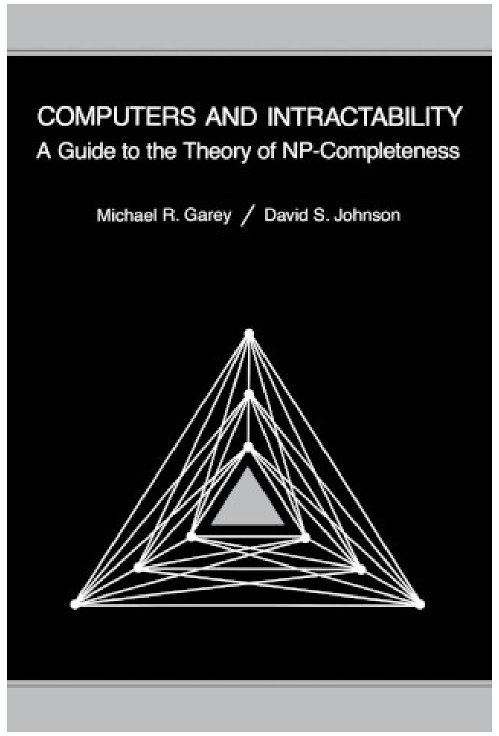
NÃO SABEMOS. Em muitas situações não sabemos nem conseguimos demonstrar matematicamente que um problema é fácil ou difícil. Temos algumas evidências que indicam, mas uma prova definitiva ainda não.

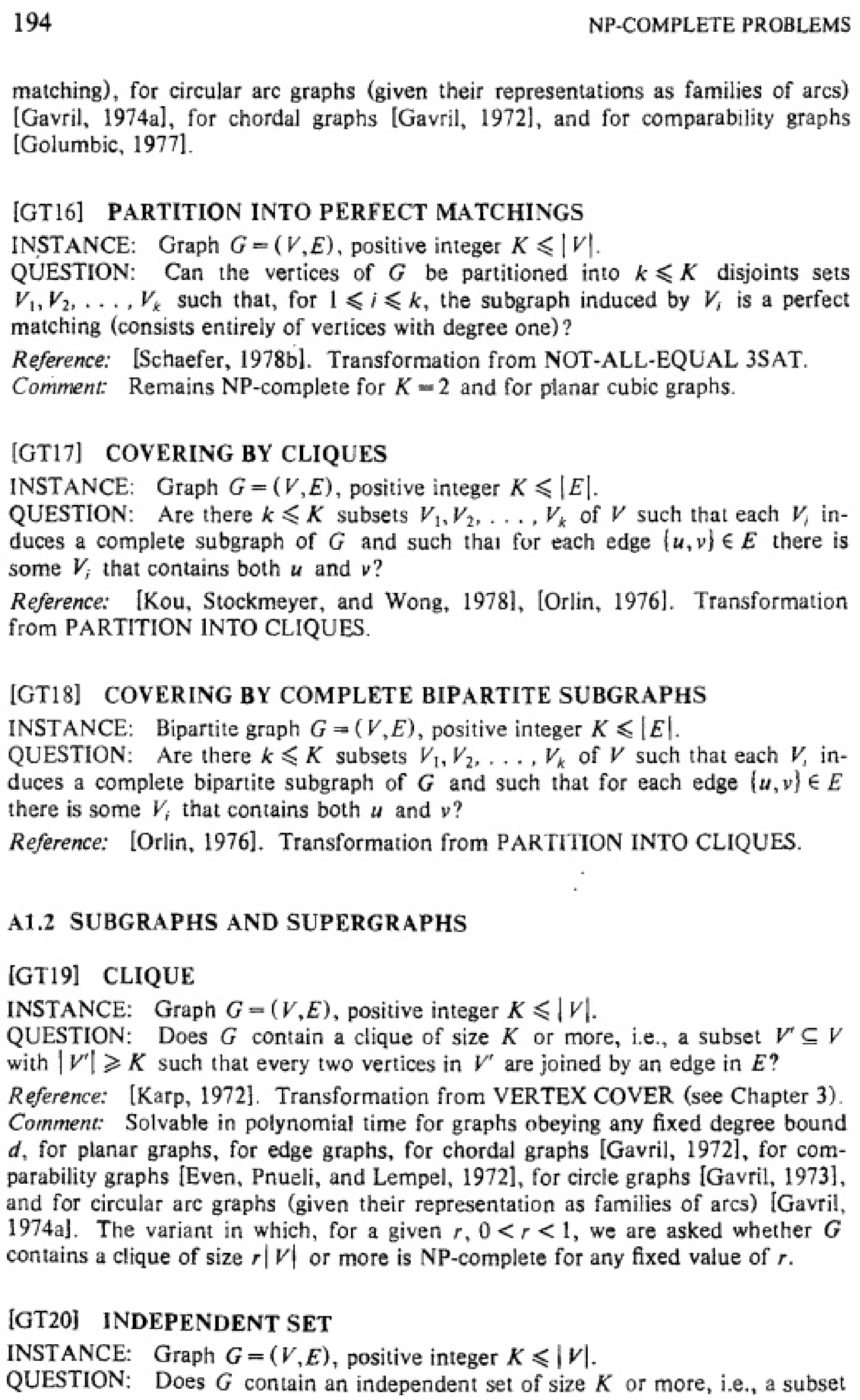
Teoria da Complexidade

Uma conquista importante dessa área foi a criação de uma "tabela periódica" para a classificação dos problemas computacionais de acordo com seu grau de dificuldade.



Teoria da Complexidade





matching), for circular arc graphs (given their representations as families of arcs) [Gavril, 1974a], for chordal graphs [Gavril, 1972], and for comparability graphs [Golumbic, 1977].

[GT16] PARTITION INTO PERFECT MATCHINGS
INSTANCE: Graph $G=(V,E)$, positive integer $K \leqslant |V|$.
QUESTION: Can the vertices of G be partitioned into $k \leqslant K$ disjoints sets $V_1, V_2, \ldots, V_k$ such that, for $1 \leqslant i \leqslant k$, the subgraph induced by V_i is a perfect matching (consists entirely of vertices with degree one)?
Reference: [Schaefer, 1978b]. Transformation from NOT-ALL-EQUAL 3SAT.
Comment: Remains NP-complete for $K=2$ and for planar cubic graphs.

[GT17] COVERING BY CLIQUES
INSTANCE: Graph $G=(V,E)$, positive integer $K \leqslant |E|$.
QUESTION: Are there $k \leqslant K$ subsets $V_1, V_2, \ldots, V_k$ of V such that each V_i induces a complete subgraph of G and such that for each edge $\{u,v\} \in E$ there is some V_i that contains both u and v?
Reference: [Kou, Stockmeyer, and Wong, 1978], [Orlin, 1976]. Transformation from PARTITION INTO CLIQUES.

[GT18] COVERING BY COMPLETE BIPARTITE SUBGRAPHS
INSTANCE: Bipartite graph $G=(V,E)$, positive integer $K \leqslant |E|$.
QUESTION: Are there $k \leqslant K$ subsets $V_1, V_2, \ldots, V_k$ of V such that each V_i induces a complete bipartite subgraph of G and such that for each edge $\{u,v\} \in E$ there is some V_i that contains both u and v?
Reference: [Orlin, 1976]. Transformation from PARTITION INTO CLIQUES.

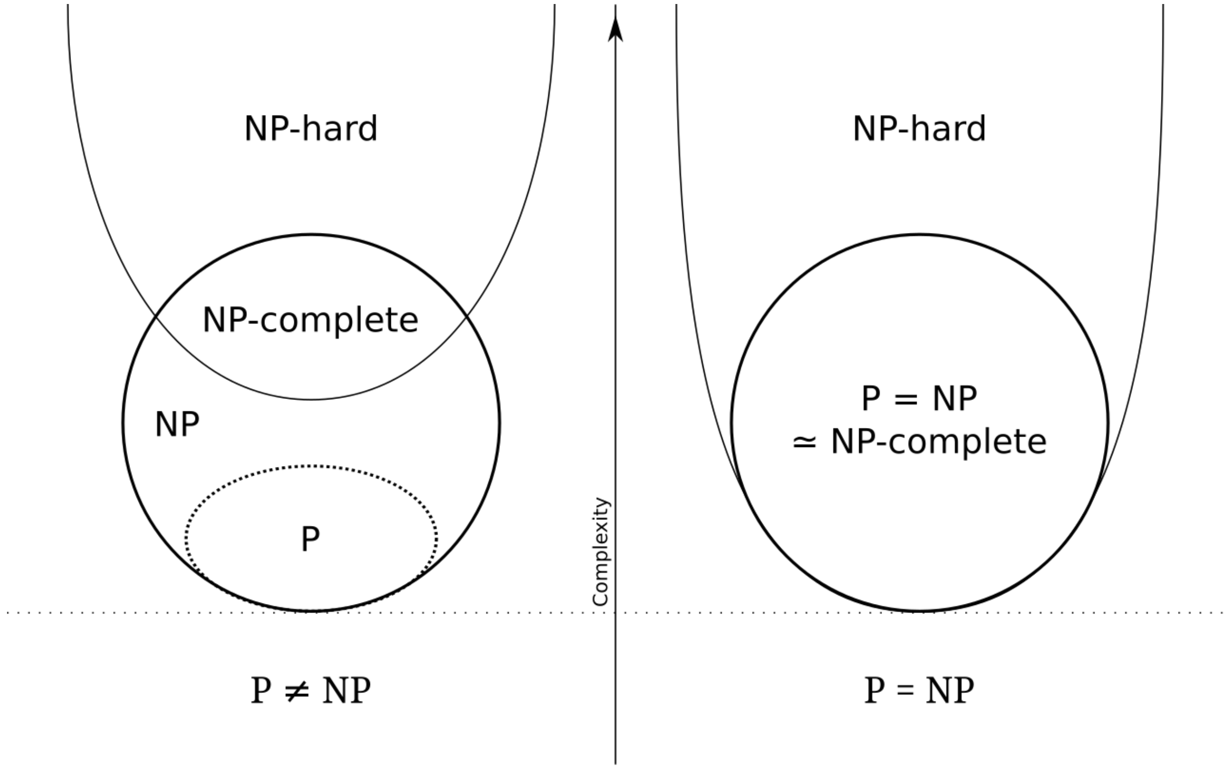
A1.2 SUBGRAPHS AND SUPERGRAPHS

[GT19] CLIQUE
INSTANCE: Graph $G=(V,E)$, positive integer $K \leqslant |V|$.
QUESTION: Does G contain a clique of size K or more, i.e., a subset $V' \subseteq V$ with $|V'| \geqslant K$ such that every two vertices in V' are joined by an edge in E?
Reference: [Karp, 1972]. Transformation from VERTEX COVER (see Chapter 3).
Comment: Solvable in polynomial time for graphs obeying any fixed degree bound d, for planar graphs, for edge graphs, for chordal graphs [Gavril, 1972], for comparability graphs [Even, Pnueli, and Lempel, 1972], for circle graphs [Gavril, 1973], and for circular arc graphs (given their representation as families of arcs) [Gavril, 1974a]. The variant in which, for a given r, $0<r<1$, we are asked whether G contains a clique of size $r|V|$ or more is NP-complete for any fixed value of r.

[GT20] INDEPENDENT SET
INSTANCE: Graph $G=(V,E)$, positive integer $K \leqslant |V|$.
QUESTION: Does G contain an independent set of size K or more, i.e., a subset

Com essas classificações conseguimos dar evidências de que um problema é difícil, mesmo que não seja possível provar.

Além disso, ainda há muita coisa que não sabemos: P = NP?



Teoria da Complexidade

Por que é importante classificar os problemas por grau de dificuldade?

→ **Alterar o problema para ficar mais fácil**

→ **Buscar uma solução não ótima (heurística)**

→ **Considerar tipos alternativos de computação**

→ **Saber quando desistir**

O objetivo é classificar os problemas em fáceis ou difíceis.

Teoria da Computabilidade

Todos os problemas podem ser resolvidos através de computadores?
Infelizmente, não.

No começo do séc. XIX, matemáticos como Kurt Gödel, Alan Turing e Alonzo Church demonstraram que existem problemas para os quais nenhuma solução pode ser encontrada por um computador. Exemplo:

→ **Verificar se uma setença matemática é verdadeira.**

→ **Como provar que 1 + 1 = 2?**

O objetivo é classificar os problemas em solucionáveis ou não.

Seja: $a = b = 1.$
Vamos provar que $2 = 1$:

$$a = b$$
$$a^2 = ab$$
$$a^2 - b^2 = ab - b^2$$
$$(a+b)(a-b) = b(a-b)$$
$$a + b = b$$
$$1 + 1 = 1$$
$$2 = 1$$

Teoria dos Autômatos

A Teoria dos Autômatos lida com as definições e propriedades matemáticas dos modelos de computação, ou seja, busca entender o modo como um computador essencialmente realiza uma computação.

Diversos modelos computacionais existem:

- **→ Autômatos finitos**
  - **Processamento de texto, compiladores, projeto de hardware**
- **→ Gramáticas livres de contexto**
  - **Linguagens de programação, inteligência artificial**
- **→ Outros ...**

O objetivo é endenter os modos de computar.

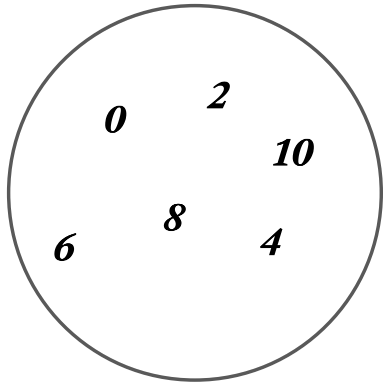
Revisão de matemática e terminologia

Conjunto:

→ **Coleção não ordenada de elementos com alguma característica em comum.**

→ **Representados por enumeração, propriedade ou diagrama.**

$$S = \{0, 2, 4, 6, 8, 10\}$$
$$S = \{x \mid (x \in \mathbb{Z}) \wedge (0 \leq x \leq 10)\}$$



→ **Se elementos repetidos importam, falamos de multiset:**

$$\{10\} \quad \{10, 10\}$$

→ **Está contido, subconjunto, subconjunto próprio:**

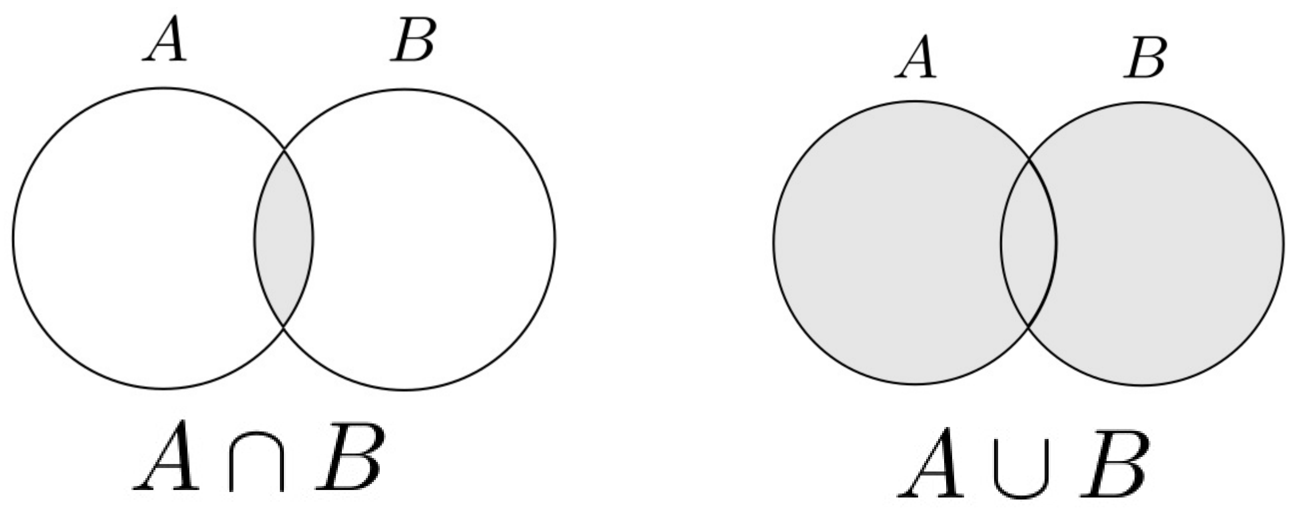
$$A \subset B \qquad B \supset A$$
$$A \subseteq B \qquad B \supseteq A$$
$$A \subsetneq B \qquad B \supsetneq A$$

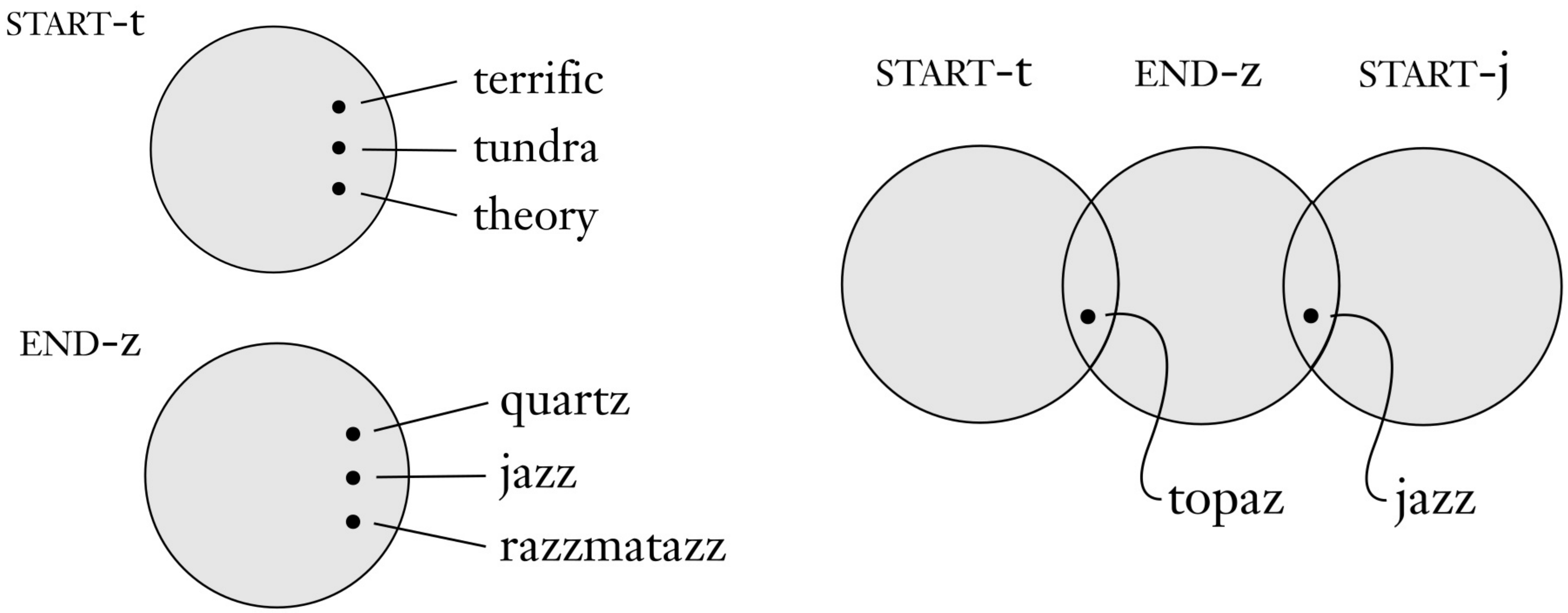
→ **Complemento:** $\overline{A}$

Revisão de matemática e terminologia

Conjunto:

→ **União e intersecção**





Revisão de matemática e terminologia

Conjunto:

→ **Produto cartesiano**

$$A \times B = \{(a, b) \mid (a \in A) \wedge (b \in B)\}$$

$$\overbrace{A \times A \times \cdots \times A}^{k} = A^k$$

→ **Número de elementos, número de subconjuntos, power set**

$$n(A) \qquad 2^{n(A)} \qquad \wp(A)$$

→ **Conjunto vazio; conjunto unitário**

$$\varnothing \text{ ou } \{\ \}$$

Revisão de matemática e terminologia

Seqüência:

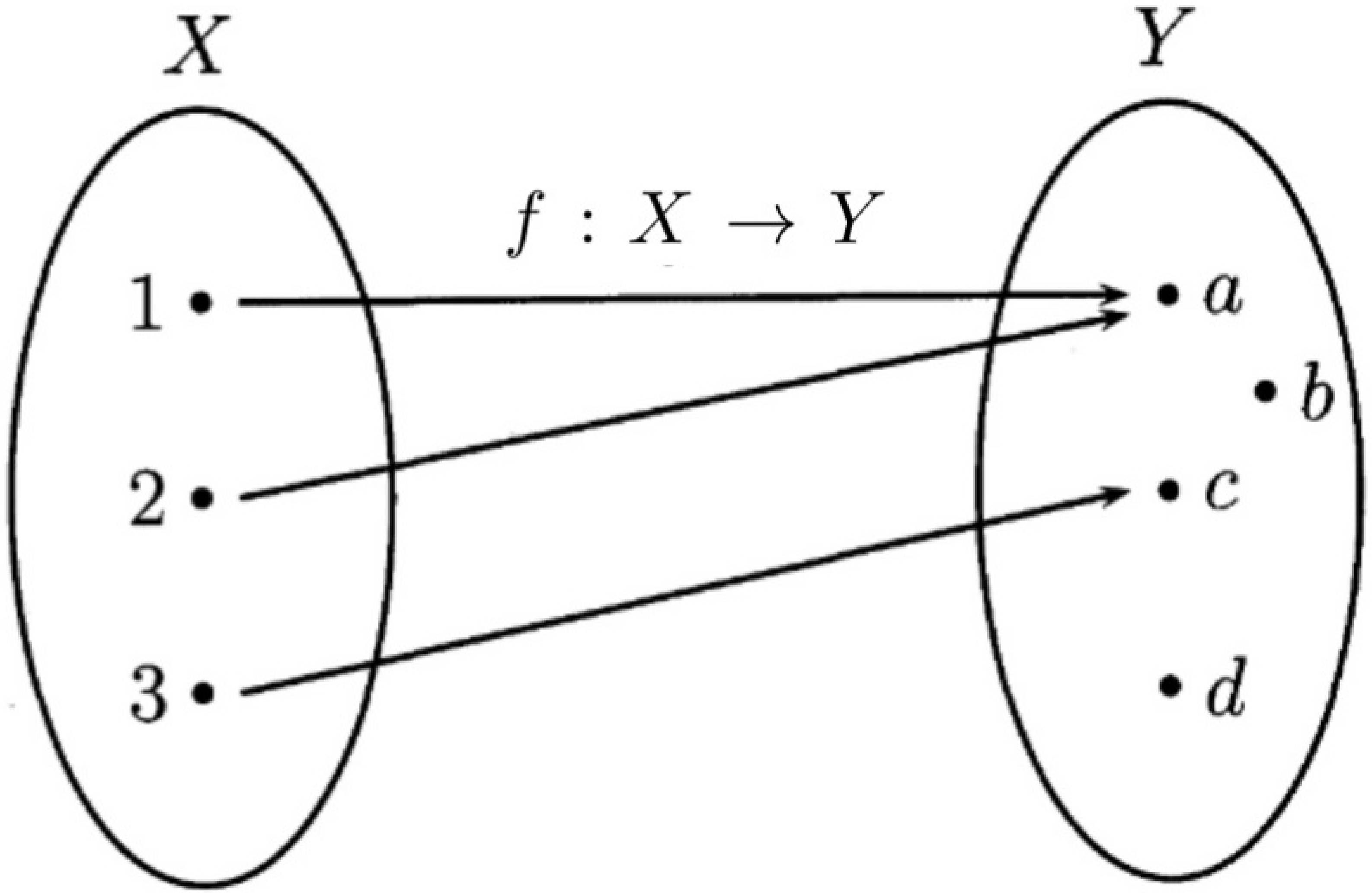
- → Uma lista de objetos em uma ordem especificada.
- → Seqüências finitas são chamadas de tuplas.
- → Uma seqüencia com k elementos é uma k-tupla.
- → Uma seqüência com 2 elementos é um par ordenado.

Revisão de matemática e terminologia

Função:

→ **É uma regra que associa a cada elemento de um conjunto X um único elemento de um conjunto Y. Todos os elementos de X devem estar associados a um único elemento de Y**

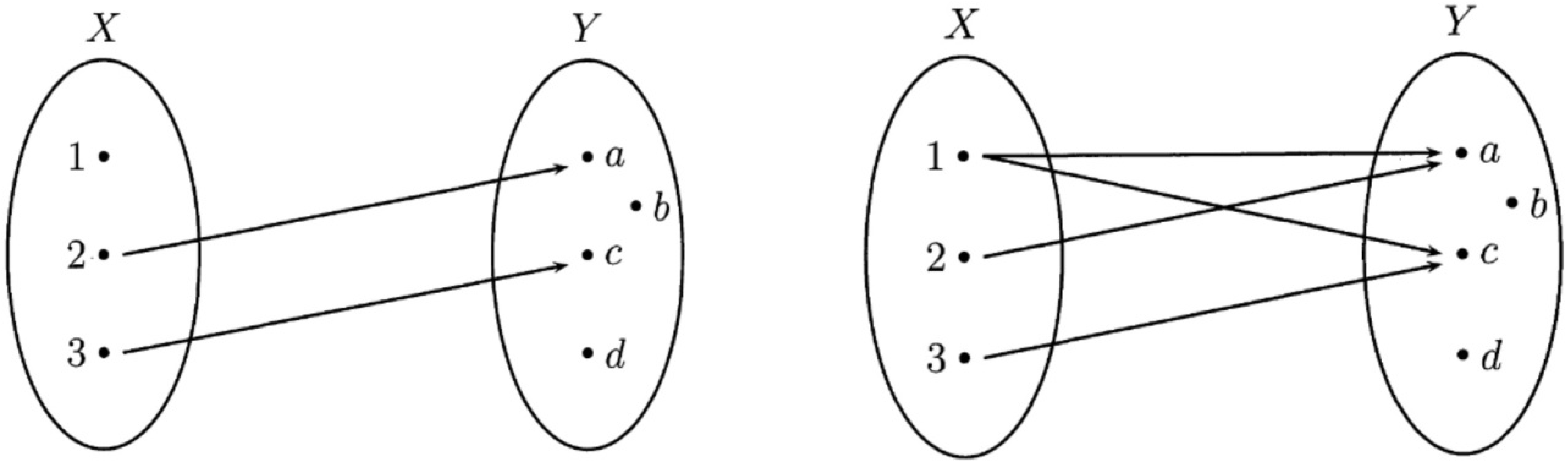
$f : X \to Y$



Revisão de matemática e terminologia

Função:

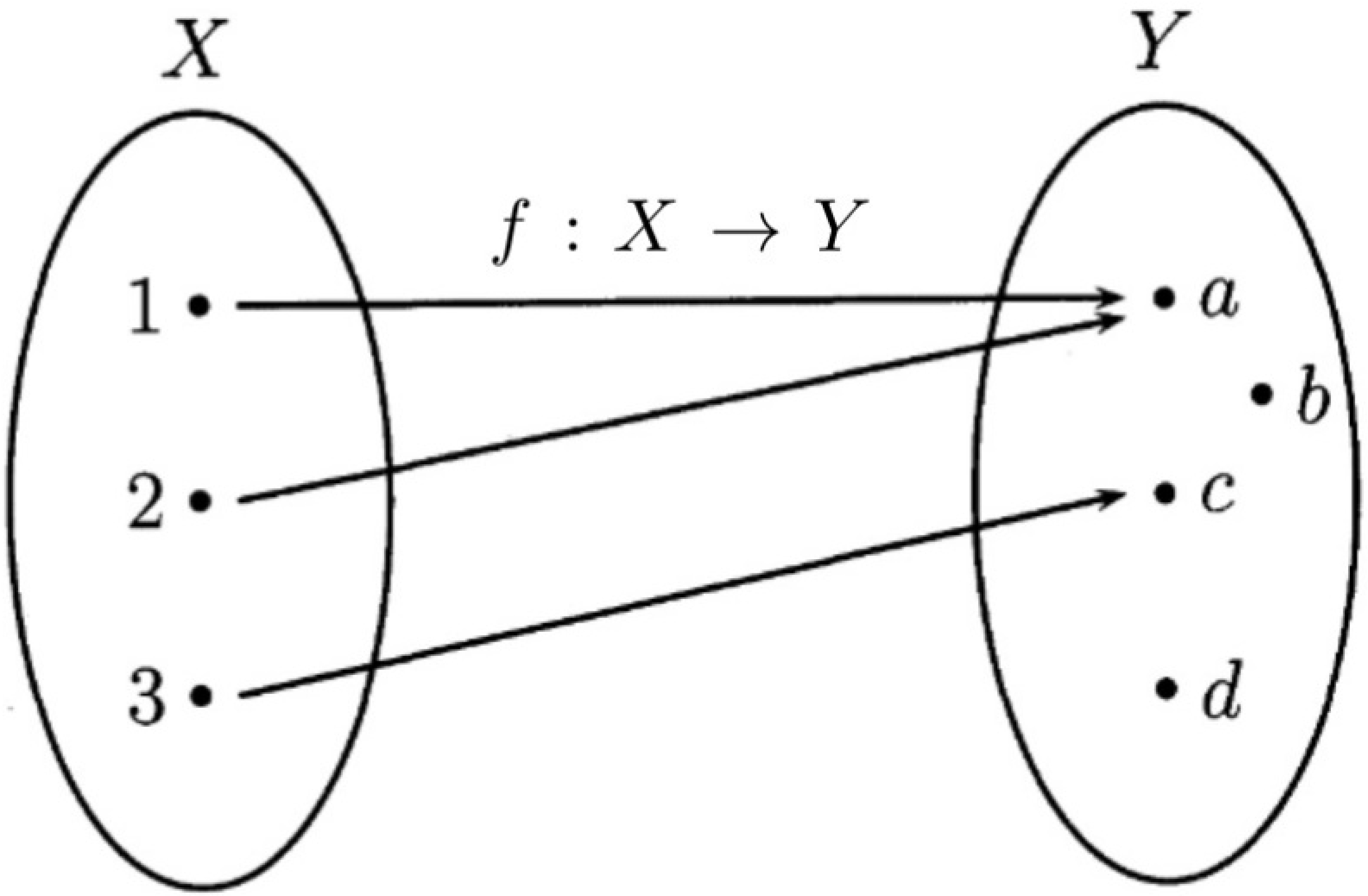
→ **Se algum elemento de X não estiver associado a um elemento de Y, ou se um elemento de X estiver associado a mais de um elemento de Y, não temos uma função.**



Revisão de matemática e terminologia

Função:

→ **O domínio de uma função são os valores de X para os quais as operações indicadas pela regra da função são possíveis. O domínio não inclui os valores de X para os quais as operações indicadas pela regra da função não são possíveis.**

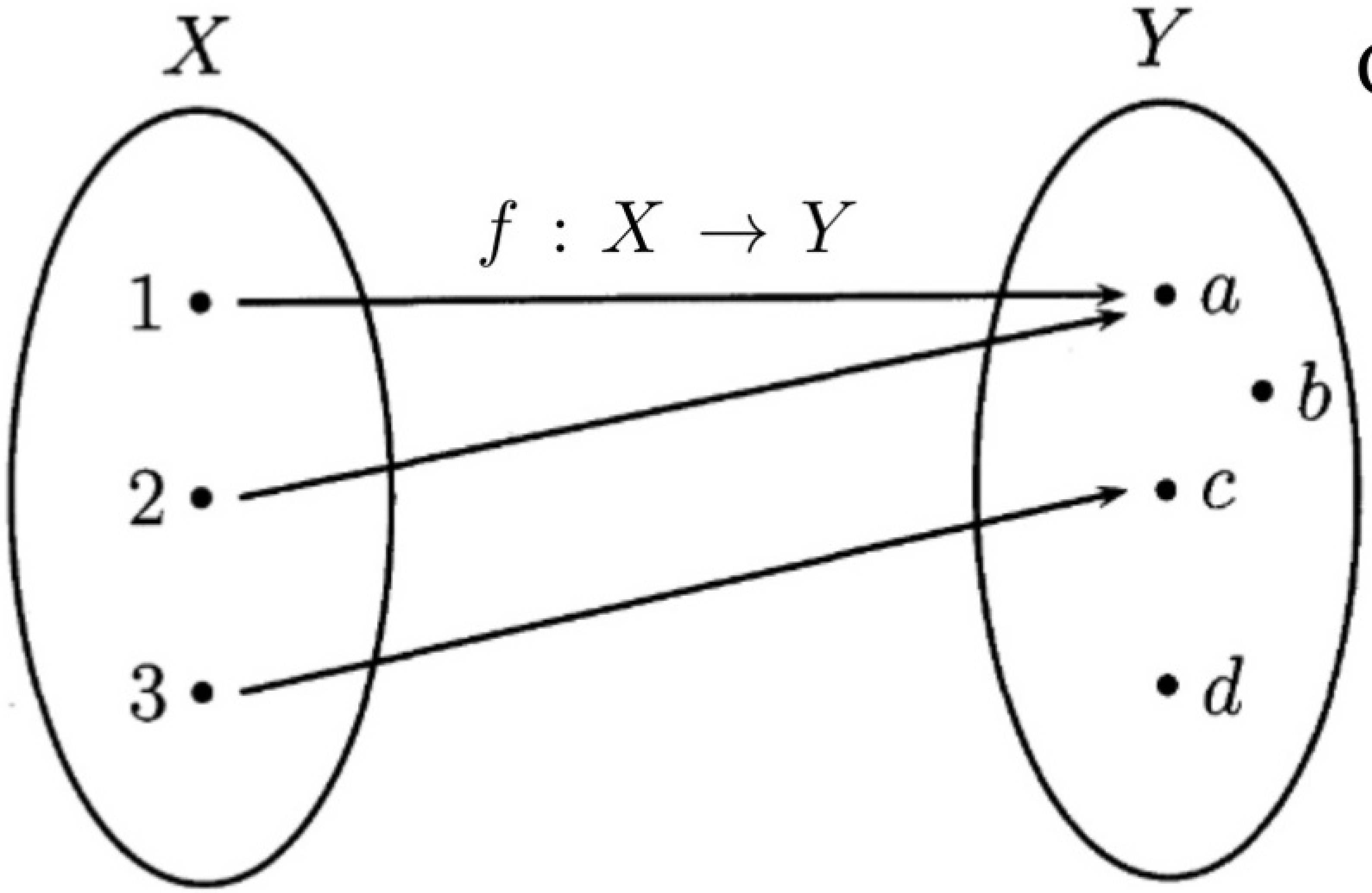


Domínio da função: $X = \{1, 2, 3\}$

Revisão de matemática e terminologia

Função:

→ **O contradomínio de uma função são todos os valores possíveis de Y. O contradomínio corresponde a tudo o que a função pode produzir como saída.**



Contradomínio da função: $Y = \{a, b, c, d\}$

Revisão de matemática e terminologia

Função:

→ **Uma imagem corresponde a um valor específico de y que está associado a um valor de x pela regra da função. Se x é um elemento de X, o único y de Y associado à x é denominado de imagem de x pela função f.**

$$y = f(x)$$

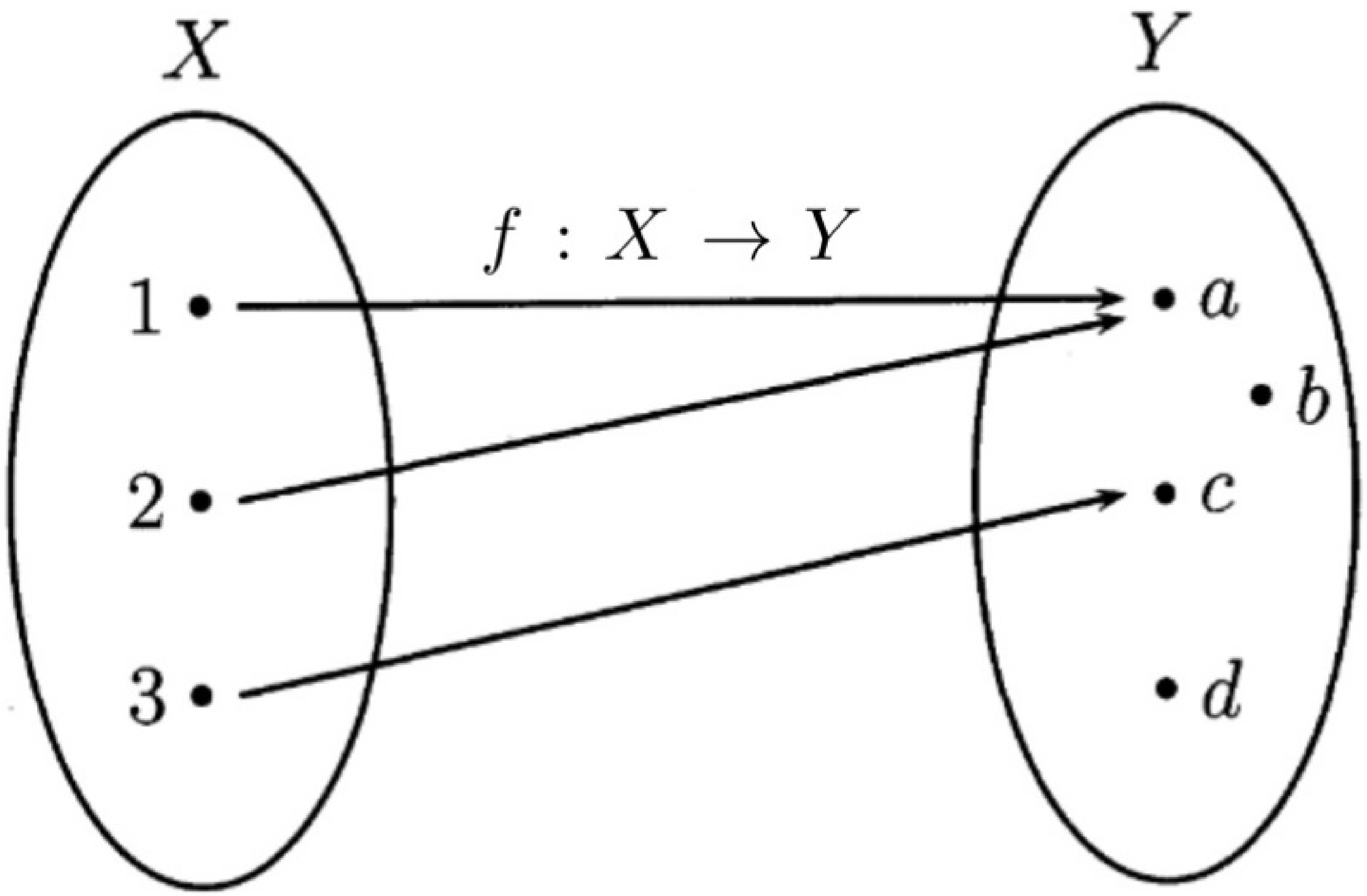


Imagem de 1: $y = f(1) = a$

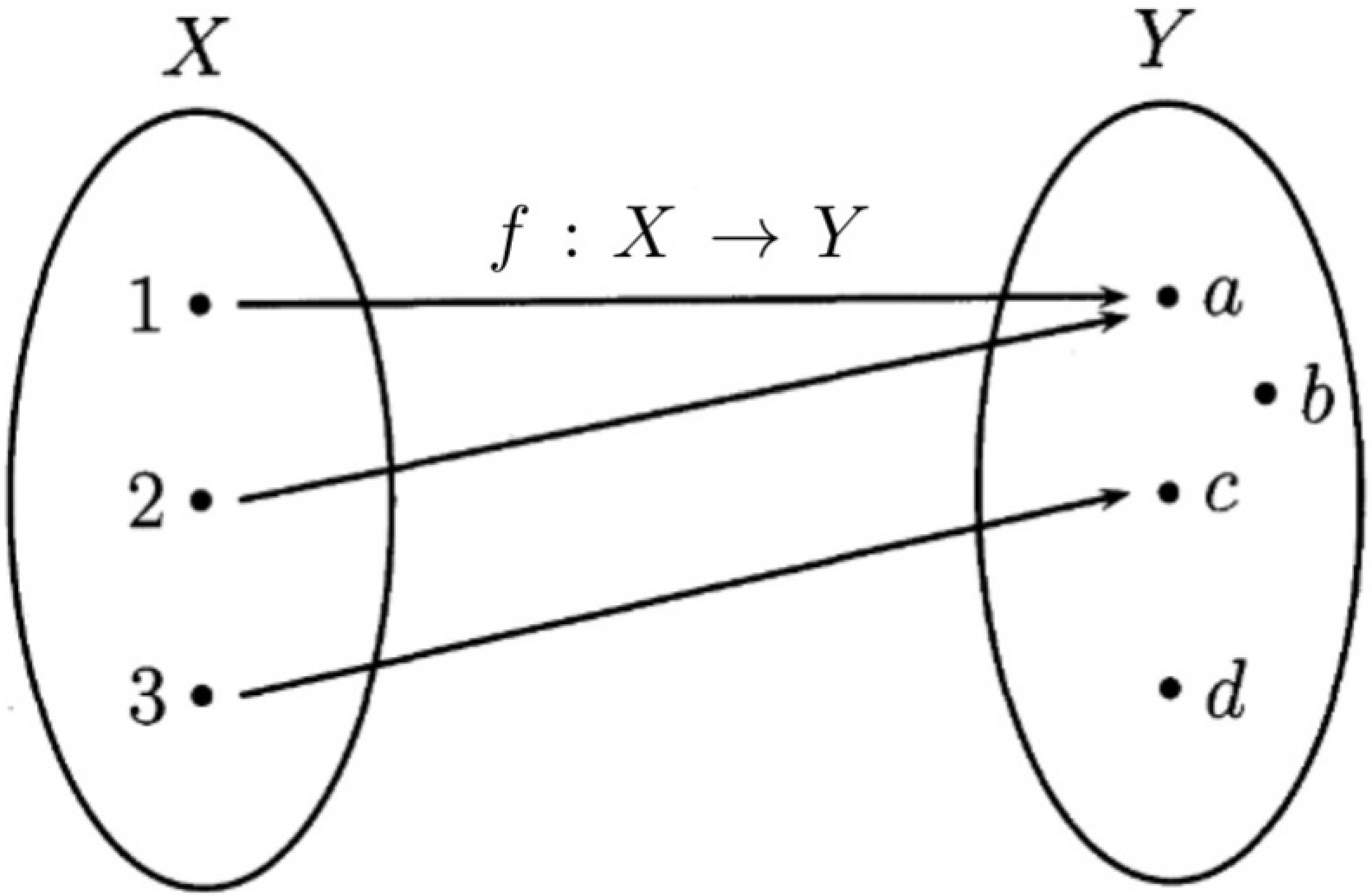
Imagem de 2: $y = f(2) = a$

Imagem de 3: $y = f(3) = c$

Revisão de matemática e terminologia

Função:

→ **O conjunto imagem corresponde ao conjunto de todas as imagens obtidas pela função.**



$$y = f(x)$$

Conjunto imagem da função: $\{a, c\}$

$$\mathrm{Im}(f) \subseteq Y$$

Revisão de matemática e terminologia

Função:

→ **Maneiras de descrever um função:**

- fórmula
- tabela de mapeamento
- gráfico
- verbalmente

→ **Quando o domínio de uma função f é dado por $A_1 \times A_2 \times \ldots \times A_k$ para alguns conjuntos $A_1, A_2, \ldots, A_k$, o input da função é a k-tupla $(a_1, a_2, \ldots, a_k)$ e nós chamamos os a_i de argumentos da função.**

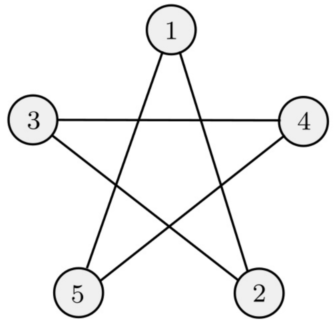
→ **Uma função com k argumentos é chamada de k-ária, e k é a aridade da função. Se k = 1 a função é unária; se k = 2 a função é binária.**

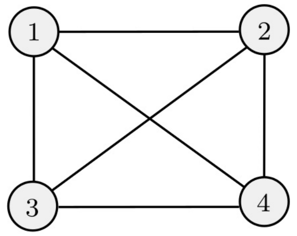
→ **Um predicado é uma função cujo contradomínio é TRUE ou FALSE.**

Revisão de matemática e terminologia

Grafos:

→ Um grafo não dirigido (ou simplesmente grafo) é um conjunto de pontos (chamados de nós ou vértices) com linhas (arestas) que conectam alguns desses pontos.





→ O número de arestas em um nó é chamado de grau desse nó. Não pode existir mais do que uma aresta entre dois nós (mas pode existir uma aresta que liga o nó nele mesmo).

→ Em um grafo G que contém os nós i e j, o par (i, j) representa a aresta que conecta esses nós. A ordem não importa em um grafo não dirigido.

→ Se V é o conjunto de nós de G, e E é o conjunto de arestas, dizemos que G = (V, E). Um grafo é descrito por um diagrama ou pela especificação de V e E:

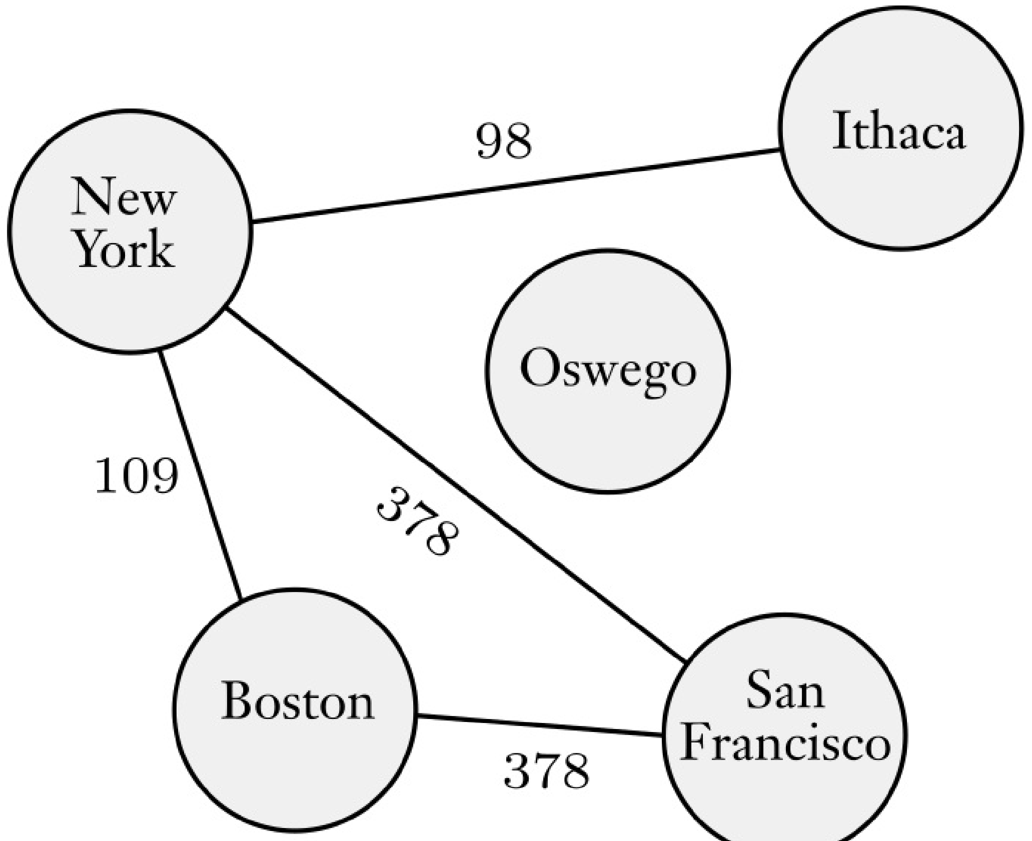
$$(\{1,2,3,4,5\},\ \{(1,2),\ (2,3),\ (3,4),\ (4,5),\ (5,1)\})$$

Revisão de matemática e terminologia

Grafos:

→ **São usados para representar dados.**

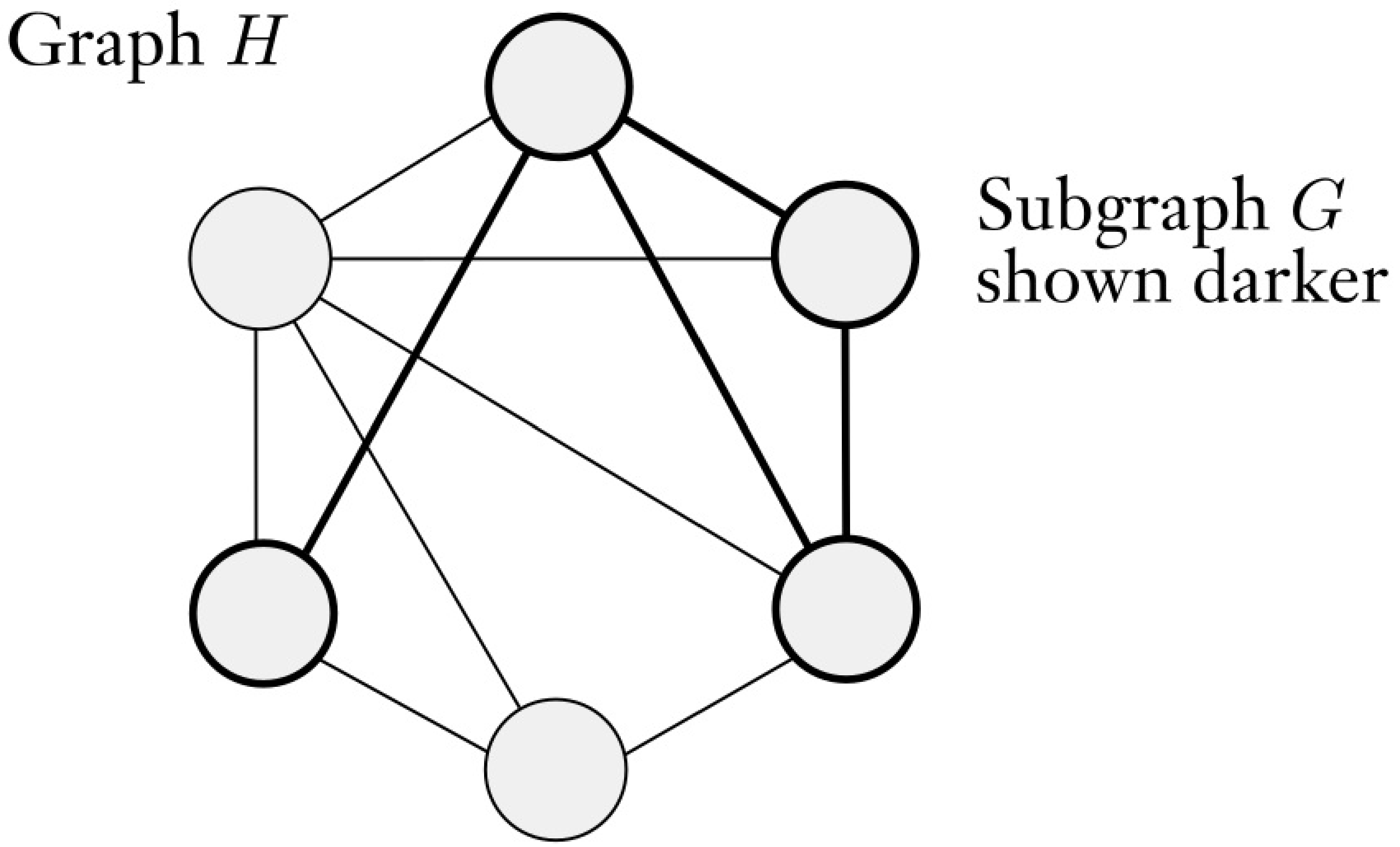
→ **Se tiverem nomes associados são chamados de grafos nomeados, etiquetados, identificados ou rotulados.**



Revisão de matemática e terminologia

Grafos:

→ Um grafo G é um subgrafo de um grafo H se os nós de G forem um subconjunto dos nós de H, e as arestas de G são as arestas de H nos nós correspondentes.



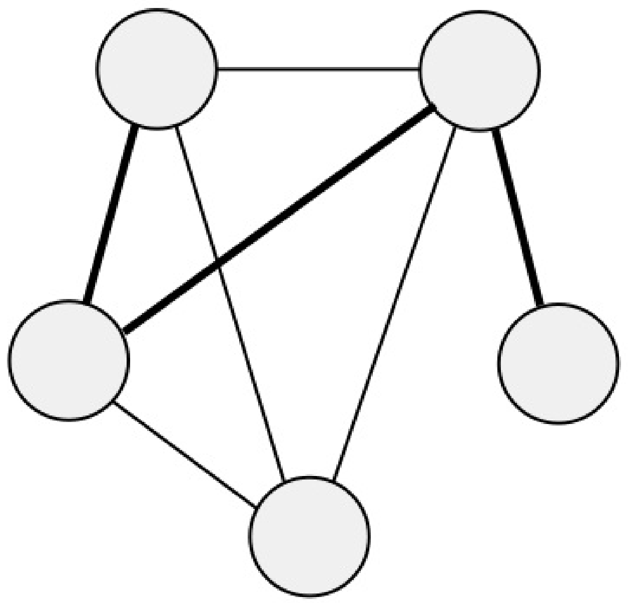
Revisão de matemática e terminologia

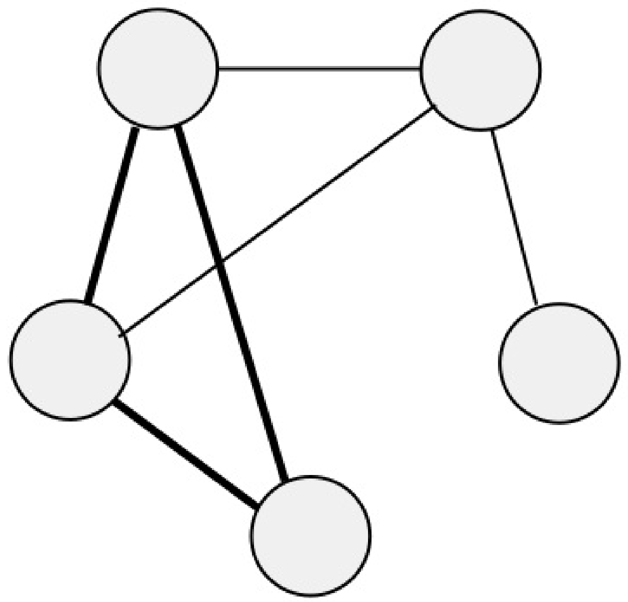
Grafos:

→ Um caminho em um grafo é uma seqüência de nós conectados por arestas. Se não houver nós repetidos, o caminho é dito simples.

→ Um grafo é conexo se existir um caminho entre todos os pares de vértices. Obs.: não é necessário que um vértice seja conectado a todos os outros!

→ Um caminho é cíclico se começar e terminar no mesmo nó. Um ciclo simples é aquele que contém pelo menos três nós e repete apenas o primeiro e o último.

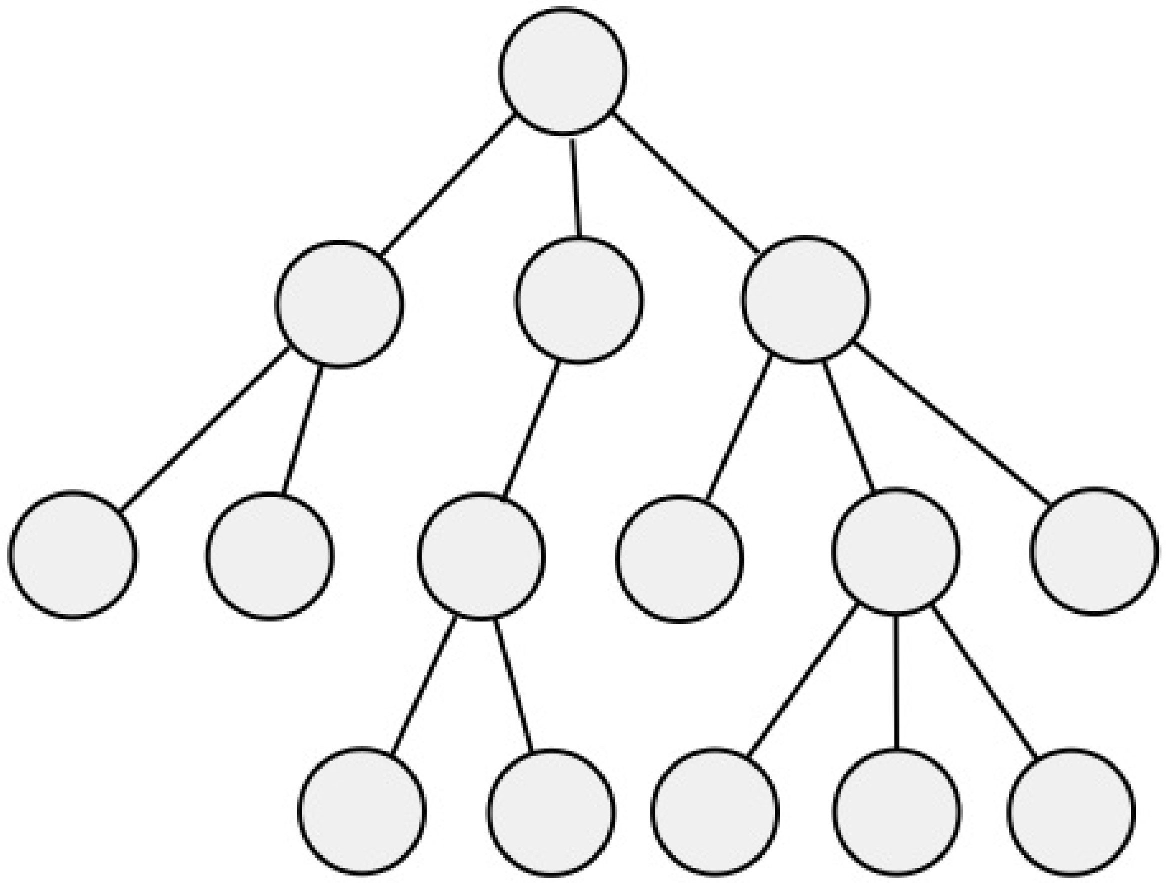




Revisão de matemática e terminologia

Grafos:

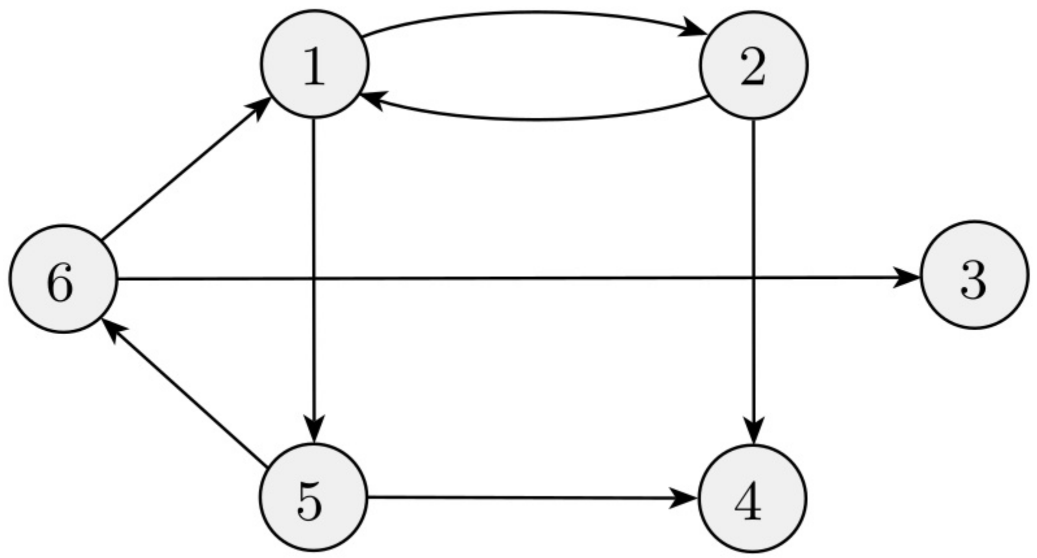
→ Um grafo é uma árvore se for conexo e não tiver ciclos simples. Um nó especial, de grau 1, pode existir e ser chamado de raiz da árvore. Os outros nós da árvore que tiverem grau 1, exceto a raiz, são chamados de folhas.

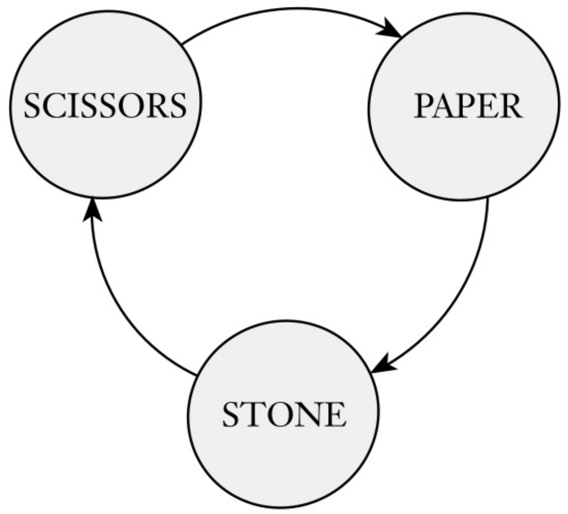


Revisão de matemática e terminologia

Grafos:

→ Um grafo é dito **dirigido** se arestas diverem uma direção.





→ O número de arestas que saem de um nó é o **grau de saída** desse nó.

→ O número de arestas que chegam em um nó é o **grau de entrada** desse nó.

→ Um grafo dirigido é **fortemente conexo** se existir um caminho entre qualquer par de nóss em ambas as direções, tal que cada nó é alcançável a partir de qualquer outro.

Revisão de matemática e terminologia

Strings e linguagens:

→ Um alfabeto é qualquer conjunto finito não vazio. Geralmente usamos as letras gregas maiúsculas Sigma ($\sum$) e Gamma (Γ) para representar um alfabeto.

$$\Sigma_1 = \{0,1\}$$

$$\Sigma_2 = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$$

$$\Gamma = \{0, 1, x, y, z\}$$

→ Os membros que pertencem a um alfabeto são seus símbolos.

→ Uma string sobre um alfabeto é uma seqüência de símbolos desse alfabeto, escritos um após o outro, não separados por vírgulas.

01001 abracadabra

Revisão de matemática e terminologia

Strings e linguagens:

→ Se w é uma string sobre Σ, o comprimento de w é escrito $|w|$ e corresponde ao número de símbolos que essa string contém.

→ A string de comprimento zero é chamada de string vazia e representada por ε.

→ Se w tem comprimento n, então $w = w_1 w_2 \cdots w_n$ para cada $w_i \in \Sigma$.

→ O reverso de w, representado por $w^{\mathcal{R}}$, é $w_n w_{n-1} \cdots w_1$.

→ A string z é substring de w se aparecer consecutivamente dentro de w.

→ Se uma string x de comprimento m e uma string y de comprimento n forem concatenadas, a concatenação de x e y é escrita como xy e é obtida colocando a string y ao final de x :

$$x_1 \cdots x_m y_1 \cdots y_n$$

Revisão de matemática e terminologia

Strings e linguagens:

→ Para concatenar uma string com ela mesma muitas vezes, usamos a notação x^k:

$$\overbrace{xx\cdots x}^{k}$$

→ A ordem lexicográfica de uma string é a mesma ordem do dicionário. Usaremos também a shortlex order (ordenação radix, lexicográfica por comprimento, militar, genealógica), que é idêncita à lexicográfica mas com string menores em primeiro lugar:

$$\Sigma_1 = \{0{,}1\} \qquad (\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots)$$

→ Uma linguagem é um conjunto de strings de um alfabeto.

Revisão de matemática e terminologia

Lógica Booleana:

→ AND, OR, NOT:

$$
\begin{array}{lll}
0 \wedge 0 = 0 & 0 \vee 0 = 0 & \neg 0 = 1 \\
0 \wedge 1 = 0 & 0 \vee 1 = 1 & \neg 1 = 0 \\
1 \wedge 0 = 0 & 1 \vee 0 = 1 & \\
1 \wedge 1 = 1 & 1 \vee 1 = 1 &
\end{array}
$$

→ XOR, equivalência, implicação:

$$
\begin{array}{lll}
0 \oplus 0 = 0 & 0 \leftrightarrow 0 = 1 & 0 \rightarrow 0 = 1 \\
0 \oplus 1 = 1 & 0 \leftrightarrow 1 = 0 & 0 \rightarrow 1 = 1 \\
1 \oplus 0 = 1 & 1 \leftrightarrow 0 = 0 & 1 \rightarrow 0 = 0 \\
1 \oplus 1 = 0 & 1 \leftrightarrow 1 = 1 & 1 \rightarrow 1 = 1
\end{array}
$$

→ Relacionamentos:

$$
\begin{array}{ll}
P \vee Q & \neg(\neg P \wedge \neg Q) \\
P \rightarrow Q & \neg P \vee Q \\
P \leftrightarrow Q & (P \rightarrow Q) \wedge (Q \rightarrow P) \\
P \oplus Q & \neg(P \leftrightarrow Q)
\end{array}
$$

→ Distributividade:

$$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$$
$$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$$

Definições, teoremas e provas

Definições:

→ Descrevem os objetos e notações que usamos. Deve ser matematicamente precisa.

Sentenças matemáticas:

→ Sentenças matemáticas sobre um objeto expressam que algum objeto tem alguma propriedade. A sentença pode ou não ser verdadeira, mas deve ser precisa.

Prova:

→ É um argumento lógico-matemático de que uma sentença matemática é verdadeira, sem qualquer dúvida ou possibilidade de dúvida.

Teorema:

→ É uma sentença matemática importante e significativa que foi provada ser verdadeira.

Lema:

→ É uma sentença menos importante que foi provada apenas como auxílio na prova de uma outra sentença mais importante e significativa.

Corolário:

→ Sentença facilmente deduzida como verdadeira a partir de um teorema.

Definições, teoremas e provas

Encontrar provas:

- → É difícil
- → Seja paciente
- → Volte ao problema várias vezes
- → Seja simples
- → Seja conciso

Definições, teoremas e provas

Tipos de provas:

→ **Construção: muitos teoremas afirmam que um tipo particular de objeto existe. Uma maneira de provar isso é demonstrar como construir um objeto.**

→ **Contradição: assumimos que um teorema é falso e, então, tentamos demonstrar que esse pressuposto é falso, ou seja, achamos uma contradição ao nosso pressuposto inicial de que o teorema é falso e, portanto, o teorema é verdadeiro.**

→ **Indução: demonstramos que todos os elmentos de um conjunto infinito tem uma propriedade específica.**

Seminários de hoje!

Chega de teoria por hoje. Vamos aos seminários preparados pelos grupos!