TEORIA DA COMPUTAÇÃO

Aula 01 – Apresentação da Disciplina e Introdução à Teoria da Computação

DIEGO KASUO NAKATA DA SILVA

Apresentação da Disciplina

- **Teoria da Computação:**

- Carga Horária Total: 68 horas
- Carga horária Semanal:

Dias da Semana	Horário
Segunda-feira	07:30h às 09:10h
Quinta-feira	09:20h às 11:00h

Apresentação da Disciplina

Objetivos

- Fornecer uma visão geral sobre as capacidades e limitações fundamentais dos computadores e sobre os problemas que podem ou não ser resolvidos computacionalmente.
- Compreender modelos computacionais, incluindo autômatos e a máquina de Turing.

Apresentação da Disciplina

Ementa

- Autômatos e Linguagens Formais, Linguagens regulares.
- Linguagens livres de contexto.
- Modelos computacionais universais, Computabilidade, Máquina de Turing, Variações da Máquina de Turing.

Apresentação da Disciplina

Material Didático

Básica:

SIPSER, Michael. **Introdução à teoria da computação**. 2.ed. São Paulo: Thomson, 2007.

HOPCROFT, John E ; ULLMAN, Jeffrey D ; MOTWANI, Rajeev. **Introdução à teoria de autômatos, linguagens e computação**. São Paulo: Campus, 2002.

LEWIS, Harry R. ; PAPADIMITRIOU, Christos H. **Elementos de teoria da computação**. Bookman, 2004.

Apresentação da Disciplina

Material Didático

Complementar:

DIVERIO, Tiaraju Asmuz; MENEZES, Paulo Fernando Blauth. **Teoria da computação: máquinas universais e computabilidade**. Bookman, 2008.

MENEZES, Paulo Fernando Blauth. **Linguagens formais e autômatos**. Bookman, 1999.

BROOKSHEAR, J. Glenn ; PIVETA, Eduardo Kessler. **Ciência da computação : uma visão abrangente**. Porto Alegre: Bookman, 2013.

Apresentação da Disciplina

Metodologia e Avaliação

Metodologia

- Aulas expositivas
- Resolução de exercícios
- Atividades individuais e coletivas (duplas)

Avaliação

- Prova escrita MP = ((P1+P2)/2)
- Frequência, participação em aula (FP) e realização das atividades.

INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Por que estudar Teoria da Computação?

O livro texto base enfoca três áreas tradicionalmente centrais da teoria da computação: **autômatos**, **computabilidade**, e **complexidade**.

- *“Quais são as capacidades e limitações fundamentais dos computadores?”*
- Começou nos anos 1930’s quando lógicos matemáticos começaram a explorar o significado de computação.

Por que estudar Teoria da Computação?

- Entender sobre as formas elementares nas quais um computador pode ser feito para pensar.
- Estabelece uma base sólida para muitas áreas abstratas da computação.
- Tenta responder as seguintes questões:
  - Quais são as propriedades matemáticas do hardware e software do computador?
  - Quais são as limitações dos computadores? Pode-se calcular "tudo"?

Por que estudar Teoria da Computação?

- Tecnologias se tornam obsoletas mas Teorias permanecem para sempre.
- Teoria da Computação provê ferramentas para resolver problemas computacionais como expressões regulares para combinações de padrões de *strings*.
- Estudar diferentes tipos de gramáticas ajuda em muitas outras áreas como projeto de compiladores e processamento de linguagem natural.

Teoria da Complexidade

- Classificar problemas de acordo com o seu grau de dificuldade.
- Dar uma prova rigorosa de que um problema que parece difícil, é verdadeiramente difícil.
- Exemplos de problemas fáceis:
  - Converter um número decimal para sua representação binária (ou vice-versa).
  - Ordenar uma sequência de 1 milhão de números. (qualquer computador resolve)
  - Computar a forma mais rápida de dirigir de uma dado lugar para outro.

Teoria da Complexidade

- Exemplos de problemas difíceis:
  - Escalonar horários de aulas e salas de aulas de todos os cursos da universidade. (1000 aulas pode levar séculos)
  - Fatorar um número inteiro de 300 dígitos em seus fatores primos.

- A questão central da teoria da complexidade é: "*O que faz alguns problemas computacionalmente difíceis e outros fáceis*?"

Teoria da Complexidade

- Comparações entre o tempo de fatoração de números com muitos algarismos em seus fatores primos, de tamanhos diferentes, quando realizadas pelos computadores atuais e por algoritmos quânticos[1].

Tamanho do número (bits)	Algoritmo Clássico	Algoritmo Quântico
512	4 dias	34 segundos
1024	100 mil anos	4.5 minutos
2048	100 mil bilhões de anos	36 minutos
4096	100 bilhões de quatrilhões de anos	4.8 horas

Teoria da Complexidade

- O Fugaku recebeu o título de computador mais rápido do mundo em 2020 após a Top 500, associação que monitora supermáquinas, ter realizado testes com o dispositivo e apontar que ele funciona até 3 vezes mais rápido que o antecessor Summit, da IBM. O supercomputador é capaz de processar 415,5 petaflops (FLoating-point Operations Per Second) por segundo. Isso quer dizer que ele pode calcular quatrilhões de contas por segundo.

Teoria da Complexidade

- Pode-se demonstrar, com um método, uma maneira de evidenciar de que certos problemas são computacionalmente difíceis.
- Quando você se depara com um problema que parece ser computacionalmente difícil, você pode:
  - Primeiro, dependendo do aspecto do problema, você pode ser capaz de alterá-lo de modo que o problema seja mais facilmente solúvel.
  - Segundo, você pode ser capaz de se contentar com menos que uma solução perfeita.
  - Terceiro, alguns problemas são difíceis somente na situação do pior caso, porém fáceis na maior parte do tempo. Dependendo da aplicação, você pode ficar satisfeito com um procedimento que ocasionalmente é lento mas resolve.
  - Finalmente, você pode considerar tipos alternativos de computação.

Teoria da Complexidade

- Na maioria das áreas, um problema computacional fácil é preferível a um difícil porque os fáceis são mais baratos de resolver.
- Criptografia é incomum porque ela especificamente requer problemas computacionais que sejam difíceis, ao invés de fáceis, porque códigos secretos têm que ser difíceis de quebrar sem a chave ou senha secreta.
- A teoria da complexidade tem mostrado aos criptógrafos o caminho dos problemas computacionalmente difíceis em torno dos quais eles têm projetado novos códigos revolucionários.

Teoria da Computabilidade

- Alan Turing delineou a ideia de computabilidade: "*Existe alguma coisa que não possa ser feita mecanicamente (sem intuição, sem inteligência)*?".
- Determinar se um enunciado matemático é verdadeiro ou falso.
- Desenvolvimento de ideias concernentes a modelos teóricos de computadores.
- A computabilidade de um problema está ligada a existência de um algoritmo que o resolva.

Teoria da Computabilidade

- Classificar problemas em resolvíveis (solúveis) e não resolvíveis (insolúveis).
- Definições formais e rigorosa das noções de computador, algoritmo e computação.
- Exemplo de problema sem solução:

➢ Hipótese de Riemann (1943): existe uma regra capaz de dizer quantos números primos existem (sequência que parece não ter lógica). Não se encontrou um meio de provar sua correção senão submetendo cada número ao teste. Isso já foi feito com os primeiros 1,5 bilhão de números e continua correta, mas ainda é pouco para se provar que ela é totalmente verdadeira.

Teoria dos Autômatos

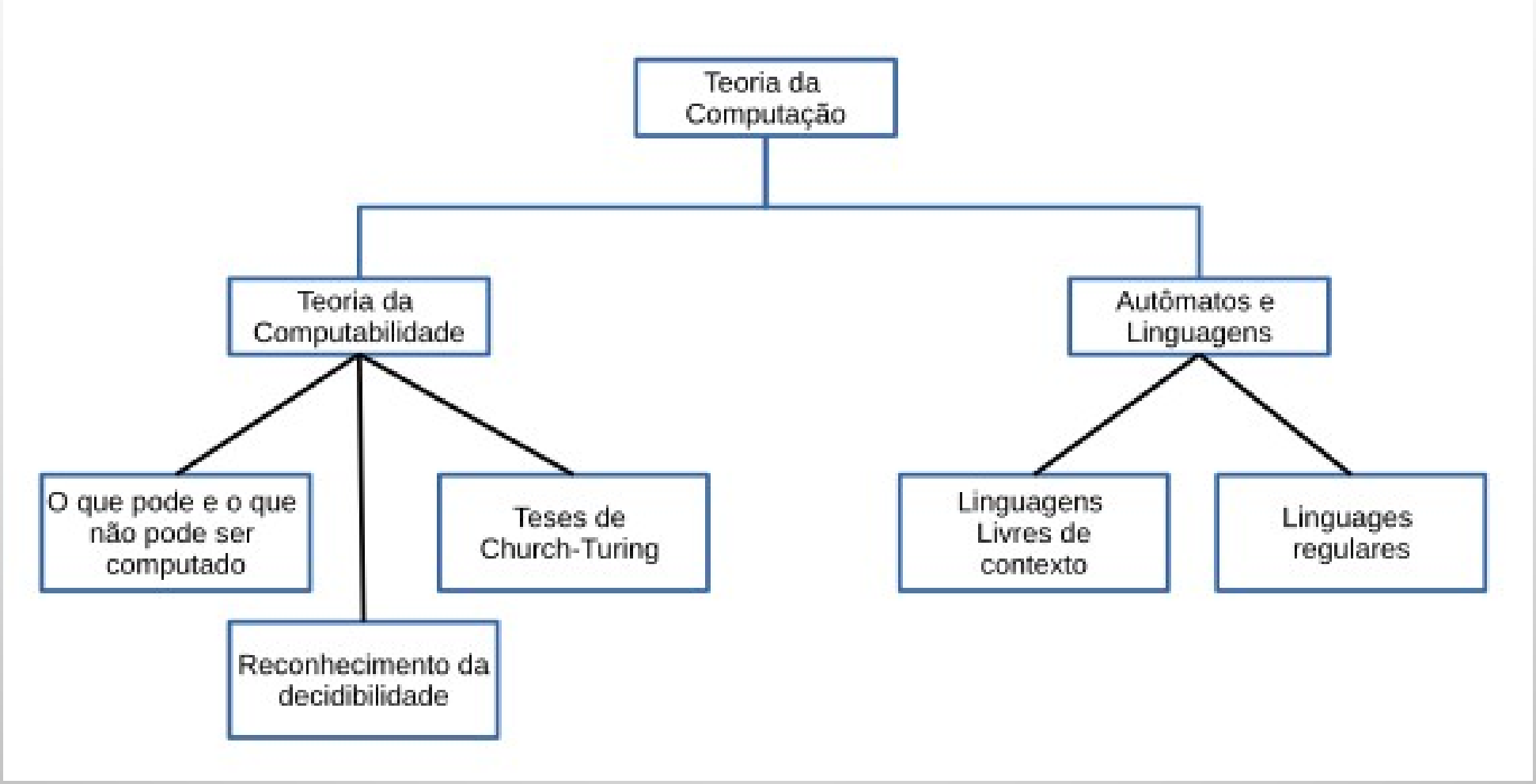
- Lida com definições e propriedades de diferentes tipos de modelos matemáticos de computação.
- Esses modelos desempenham um papel em diversas áreas aplicadas da ciência da computação.
- Exemplo de modelos:
  - **Autômatos finitos**: modelo usado em processamento de texto, compiladores e projeto de hardware.
  - **Gramática livre de contexto**: modelo usado em linguagens de programação e inteligência artificial.
  - **Máquina de Turing**: modelo abstrato simples de um computador real como o que utilizamos hoje em dia (nosso PC é uma máquina de Turing com memória limitada).

Relação entre as áreas da Teoria da Computação

Em resumo:
Na Teoria da Complexidade, o objetivo é classificar problemas como fáceis ou difíceis, enquanto que na Teoria da Computabilidade a classificação de problemas é por meio da separação entre os que são solúveis e os que não são. Ambas requerem uma definição precisa de um computador.
A Teoria dos Autômatos permite praticar com definições formais de computação pois ela introduz conceitos relevantes a áreas não teóricas da computação.

- Faremos um estudo das duas últimas áreas, na ordem reversa. Começaremos com Teoria dos Autômatos e seguiremos com Teoria da Computabilidade.

Relação entre as áreas da Teoria da Computação



NOÇÕES E TERMINOLOGIA MATEMÁTICAS

Noções e Terminologias Matemáticas

- Conjuntos
- Sequências e Uplas
- Funções e Relações
- Grafos
- Cadeias e Linguagens
- Lógica Booleana

Noções e Terminologias Matemáticas

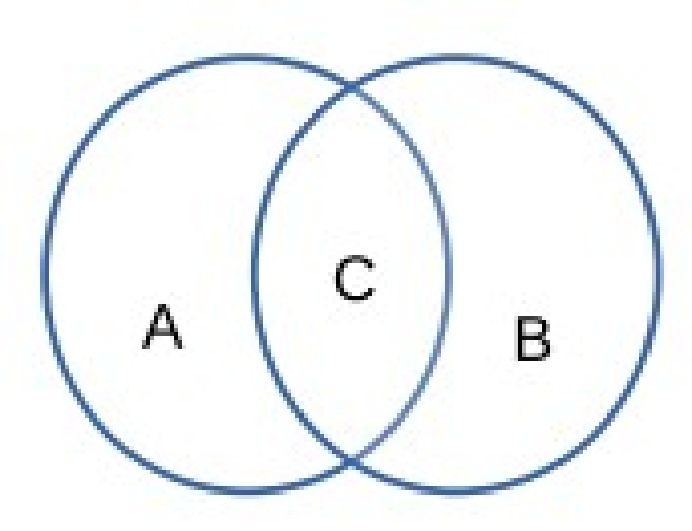
- Conjuntos
- Sequências e Uplas
- Funções e Relações
- Grafos
- Cadeias e Linguagens
- Lógica Booleana

Conjuntos

- Um **conjunto** é um grupo de objetos representado como uma unidade.
- Os objetos em um conjunto são chamados **elementos** ou **membros**.

Conjuntos

- Descrição de conjuntos:
  - Listar elementos entre chaves:
  - Descrever com uma regra:
  - Usando diagrama de Venn:



Conjuntos

- Descrição de conjuntos:
  - A ordem de descrever um conjunto não importa, nem a repetição de seus membros.

    $\{ 57, 7, 7, 7, 21 \} = \{ 57, 7, 21 \}$.
  - Se desejamos levar em consideração o numero de ocorrências de membros chamamos o grupo um ***multiconjunto*** ao invés de um conjunto. Logo $\{ 7 \}$ e $\{ 7, 7 \}$ são diferentes como multiconjunto mas idênticos como conjuntos.
  - Um conjunto infinito contém uma quantidade infinita de elementos. O conjunto de números naturais N, $\{ 1, 2, 3, \ldots \}$. O conjunto de inteiros Z, escrito como $\{ \ldots, -2, -1, 0, 1, 2, \ldots \}$.
  - O conjunto com 0 membros é chamado o conjunto vazio e é escrito $\varnothing$.

Conjuntos

Relações e Operações sobre Conjuntos

1. Pertinência:

- Pertence a:
- Não pertence a:

Conjuntos

Relações e Operações sobre Conjuntos

2. Inclusão:

- É subconjunto: $\{1,18\} \subset \{1,2,18\}$
- Não é subconjunto: $\{1,19\} \not\subset \{1,2,18\}$
- É superconjunto : $\{1,2,18\} \supset \{1,18\}$
- Não é superconjunto: $\{1,2,18\} \not\supset \{1,10\}$
- É subconjunto: $\{1,2,18\} \subseteq \{1,2,18\}$
- Não é subconjunto: $\{1,3,19\} \not\subseteq \{1,2,18\}$
- É superconjunto: $\{1,2,18\} \supseteq \{1,2,18\}$
- Não é superconjunto: $\{1,2,19\} \not\supseteq \{1,2,18\}$

Conjuntos

Relações e Operações sobre Conjuntos

- União:
- Interseção:
- Complementar:
- Diferença: ou

Conjuntos – Exercício

Examine as descrições formais de conjuntos abaixo de modo que você entenda quais membros eles contem. Escreva uma descrição informal breve em português de cada conjunto.

Conjuntos – Exercício

Escreva descrições formais dos seguinte conjuntos:

1. O conjunto contendo os números 1, 10, e 100.
2. O conjunto contendo todos os inteiros que são maiores que 5.
3. O conjunto contendo todos os naturais que são menores que 5.

Conjuntos – Exercício

Seja e :

1. A é um subconjunto de B?
2. B é subconjunto de A?
3. Quem é A∪B ?
4. Quem é A∩B ?
5. Quem é o complementar de A em relação a B?
6. Quem é o complementar de B em relação a A?.

Sequências e Uplas

- Uma **sequência de objetos** é uma lista desses objetos na mesma ordem.
- Em um conjunto a **ordem** e **repetição** não importam, mas em sequências importam.
- **Uplas** são sequências finitas. Uma sequência com *k* elementos é uma ***k-upla***.

Sequências e Uplas

Exemplos:

- A sequência 7, 21, 57 pode ser escrita como uma k-upla
- (7, 21, 57) com , ou seja .
- (7, 21, 57) não é o mesmo que (7, 57, 21), e ambas são diferentes de (7, 7, 21, 57).
- {7, 21, 57} é idêntico a {7, 7, 21, 57}.

Sequências e Uplas

- Conjuntos e sequências podem aparecer como elementos de outros conjuntos e sequências.
- O **conjunto das partes** de um conjunto A é um conjunto de todos os subconjuntos de A.

Sequências e Uplas

Exemplos:

Seja

- O conjunto das partes de B é o conjunto .
- O conjunto de todos os pares cujos elementos são 0s e 1s é .

Sequências e Uplas

Se A e B são dois conjuntos, o **produto cartesiano** ou **produto cruzado** de e , escrito como , é o conjunto de todos os pares nos quais o primeiro elemento é membro de A e o segundo elemento é um membro de B.

Sequências e Uplas

Exemplos:

Seja e .

- .

Funções e Relações

- Uma **função** ou **mapeamento** é um objeto que estabelece um relacionamento entrada-saída.
- É um procedimento para computar uma saída a partir de uma entrada especificada.
- Se f é uma função cujo valor de saída é b quando o valor de entrada é a, escrevemos **f(a) = b**, onde f mapeia a para b.
- O conjunto das possíveis entradas para a função é chamado **domínio**. As saídas da função vêm de um conjunto chamado **contradomínio**.

Funções e Relações

- A notação representa que "f é uma função com domínio em e contradomínio em ".
- Uma função pode não necessariamente usar todos os elementos do contradomínio especificado.
- Uma função que usa todos os elementos do contradomínio é dita ser **sobre o contradomínio**.

Funções e Relações

Exemplos:

- Função do valor absoluto: e .
- Na função , se o domínio e o contradomínio é o conjunto $\mathbb{Z}$, então .
- Função da adição: e
- Na função , o domínio é o conjunto de pares de inteiros e o contradomínio é , então .

Funções e Relações

- Quando o domínio de uma função é para alguns conjuntos a entrada para é uma e chamamos os de **argumentos**.
- Uma função com argumentos é chamada função , e é dita a ***aridade*** da função. Se é 1, tem um único argumento e é chamada uma ***função unária***. Se é 2, é uma ***função binária***.

Funções e Relações

- Funções binárias familiares são escritas em uma notação ***infixa*** (símbolo da função entre argumentos) ao invés da notação ***prefixa*** (símbolo da função precedendo os argumentos): ao invés de , por exemplo.
- Um ***predicado*** ou ***propriedade*** é uma função cujo contradomínio é . Ex.:

Funções e Relações

Exemplo:

- Em um jogo infantil chamado Tesoura–Papel–Pedra, os dois jogadores escolhem simultaneamente um membro do conjunto { TESOURA, PAPEL, PEDRA } e indicam suas escolhas com sinais de mão. Se as duas escolhas são iguais, o jogo começa. Se as escolhas diferem, um jogador vence, conforme a relação *bate*.

bate	TESOURA	PAPEL	PEDRA
TESOURA	FALSO	VERDADEIRO	FALSO
PAPEL	FALSO	FALSO	VERDADEIRO
PEDRA	VERDADEIRO	FALSO	FALSO

Dessa tabela determinamos que TESOURA bate PAPEL é VERDADEIRO e que PAPEL bate TESOURA é FALSO.

Funções e Relações

Um tipo especial de relação binária é chamada um ***relação de equivalência R*** se R satisfaz três condições:

- 1. é **reflexiva** se para todo , ;
- 2. é **simétrica** se para todo e , implica ; e
- 3. é **transitiva** se para todo , , e , e implica .

Grafos

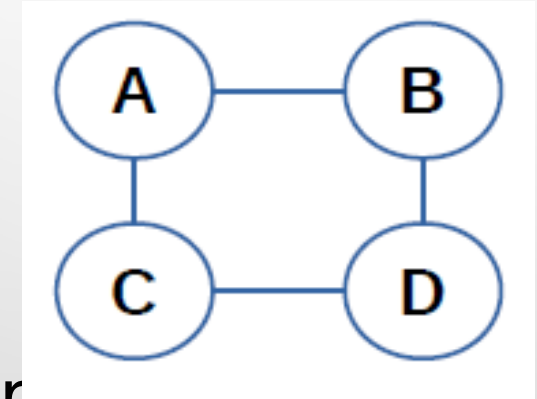
- Um ***grafo não-direcionado***, ou simplesmente um ***grafo*** (representado por) é um conjunto de pontos com linhas conectando alguns dos pontos.
- Os pontos são chamados **nós (vértices, V)** e as linhas são arestas **E**.
- O número de arestas de um dados nó representa o **grau** desse nó.

Grafos

Exemplos:

- Na figura, o tem e

.



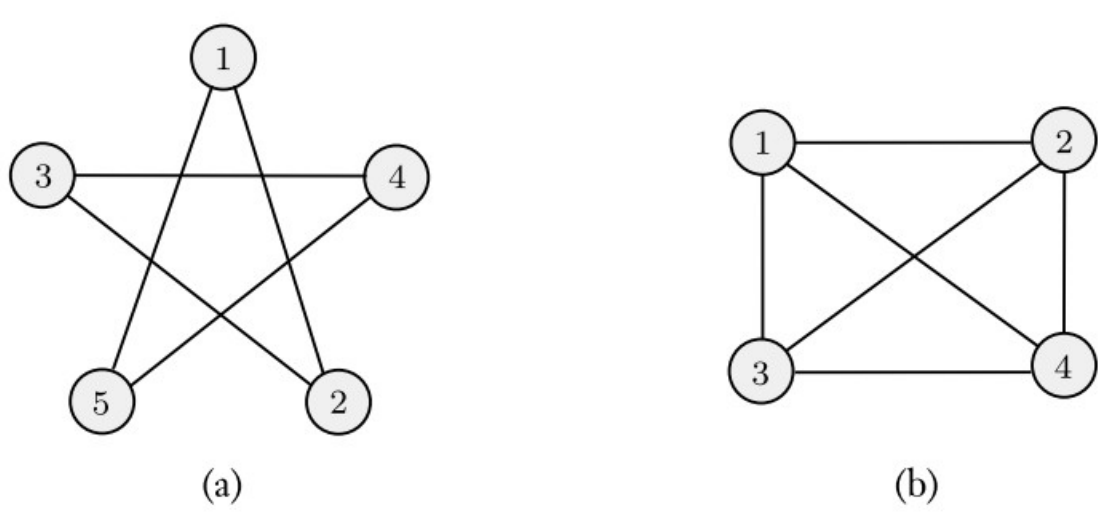
Aqui, a ordem das coordenadas das arestas não importa.

*Grafos frequentemente são usados para representar dados. Nós podem ser cidades e arestas as estradas que as conectam, ou nós podem ser componentes elétricas e arestas os fios entre elas

Grafos

Exemplos:

- Seja

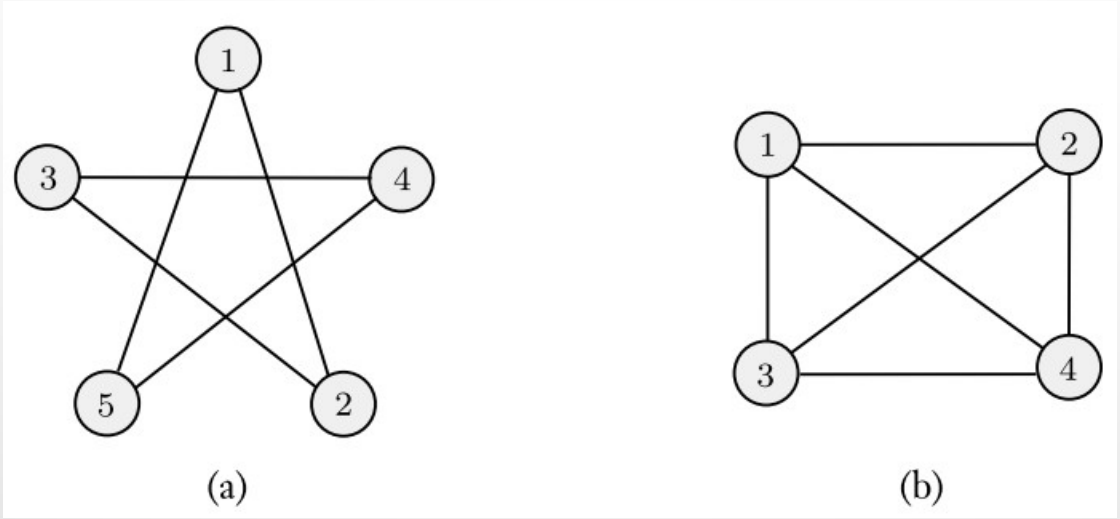


- O **número de arestas** em um nó específico é o **grau daquele nó**. Na Figura acima, (a) todos os nós têm grau 2. Na Figura (b) todos os nós têm grau 3.
- Não mais que uma aresta é permitida entre quaisquer dois nós.

Grafos

Exemplos:

- Seja



uma descrição formal do grafo na Figura (a) é

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

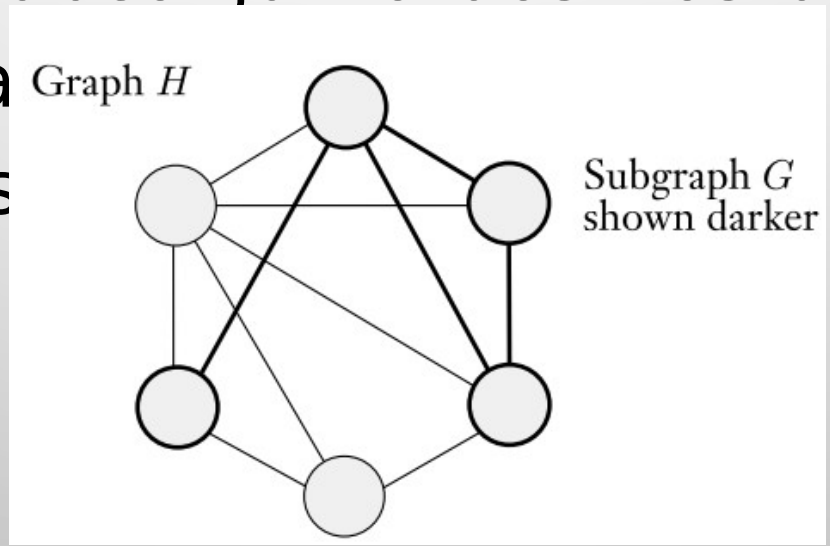
e uma descrição formal do grafo na Figura (b) é

({ 1, 2, 3, 4 } , { (1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4) }).

Grafos

Definições:

- **Grafos rotulados** possui identificação em vértices e arestas.
- Um grafo G é um ***subgrafo*** de um grafo H se os nós de G formam um subconjunto dos nós de H, e as arestas de G são a sobre os nós correspondentes



Grafos

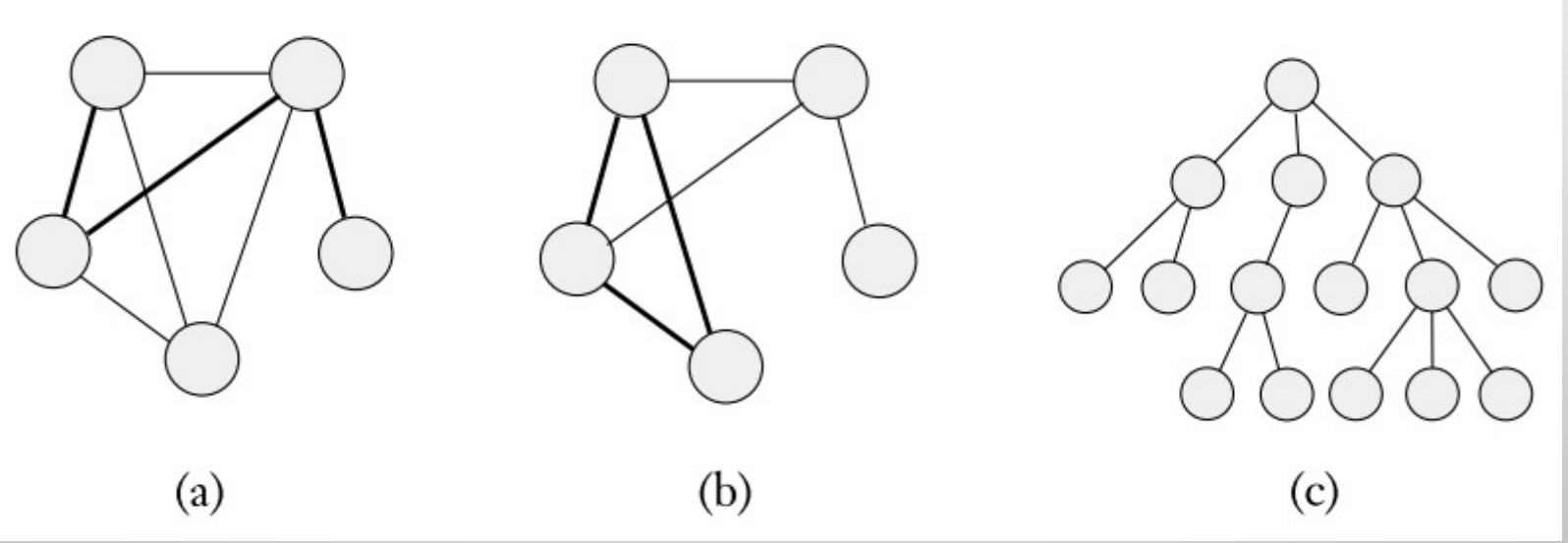
Definições:

- Um **caminho** em um grafo é uma sequência de nós conectados por arestas.
- Um **caminho simples** é um caminho que não repete nenhum nó.
- Um grafo é **conexo** se cada dois nós têm um caminho entre eles.
- Um caminho é um **ciclo** se ele começa e termina no mesmo nó
- Um **ciclo simples** é aquele que contém pelo menos três nós e repete somente o primeiro e último nós.

Grafos

Definições:

- **Árvore** é um grafo conexo sem ciclos.
- **Raiz** é o nó de origem da árvore
- Os nós de grau 1 em uma árvore, exceto a raiz, são chamados as **folhas** da árvore.

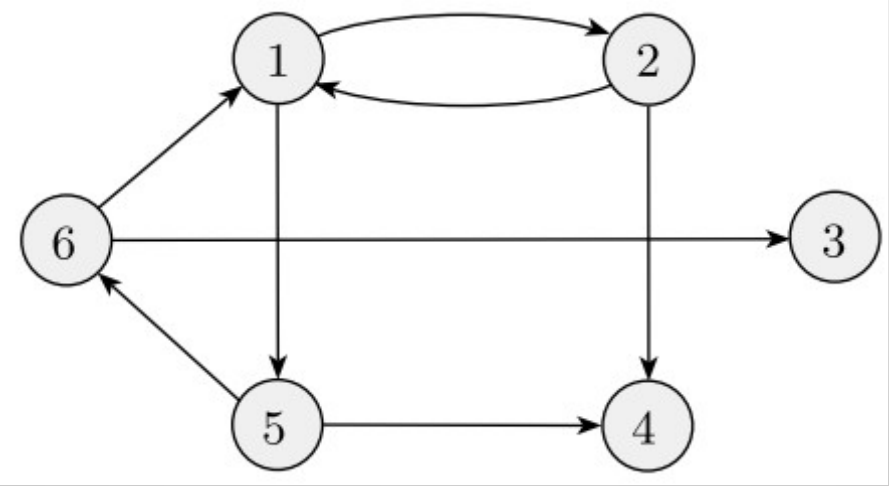


(a) Um caminho em um grafo, (b) um ciclo em um grafo, e (c) uma árvore

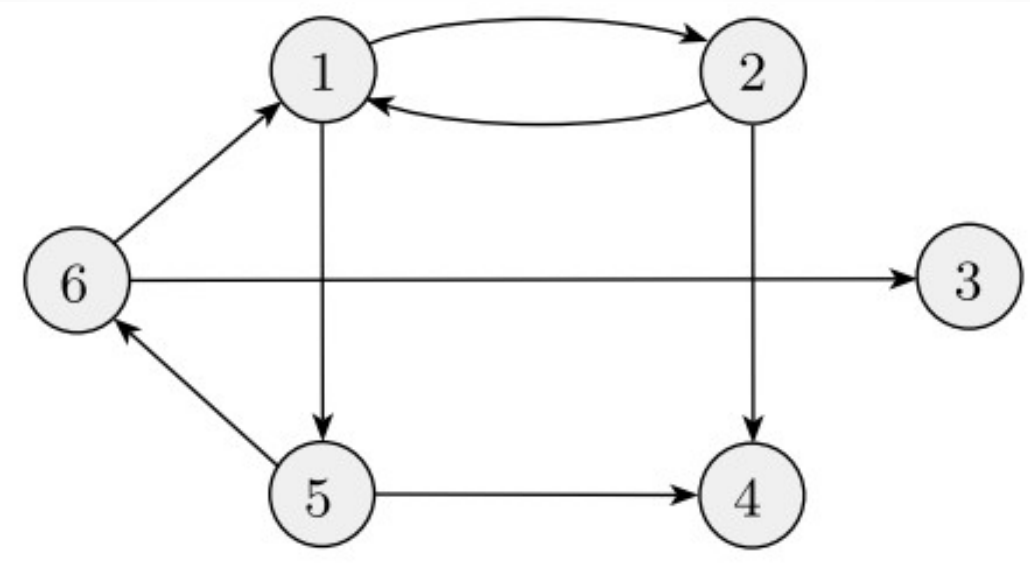
Grafos

Definições:

- Um grafo com “setas” ao invés de linhas é um **grafo direcionado**.
- O número de setas apontando para um dado nó representa o **grau de entrada.**
- O número de setas apontando a partir de um dado nó representa o **grau de saída.**
- Um grafo no qual t a direção de seus passos é chamado um **cam**



Grafos



Em um grafo direcionado representamos uma aresta de para como um par . A descrição formal de um grafo direcionado é onde é o conjunto de nós e é o conjunto de arestas. A descrição formal do grafo acima é

({1,2,3,4,5,6 } , { (1,2), (1,5), (2,1), (2,4), (5,4), (5,6), (6,1), (6,3) })

Cadeias e Linguagens

- **Alfabeto** é um conjunto finito de caracteres usados na escrita de uma linguagem.
- Os membros do alfabeto são os **símbolos (caracteres)** do alfabeto.
- Geralmente usamos letras gregas maiúsculas para designar alfabetos.
- Também usamos as letras minúsculas do nosso alfabeto para representar os símbolos.

Cadeias e Linguagens

Exemplos:

- .
- .

Cadeias e Linguagens

- Uma **cadeia sobre um alfabeto (palavra)** é uma sequência de símbolos daquele alfabeto, usualmente escrito um seguido do outro e não separados por vírgulas.
- O **comprimento** de uma cadeia é o número de símbolos que ela contém. Se é uma cadeia, então seu comprimento é representado por .
- Uma **cadeia vazia** é uma cadeia de comprimento zero, e é representada por (Épsilon). Uma cadeia sem símbolos é uma cadeia válida.
- O **reverso** de uma cadeia é essa cadeia escrita na ordem inversa, é representado por

Cadeias e Linguagens

Exemplos:

- pode gerar a cadeia .
- O comprimento de é 5, isto é,

Cadeias e Linguagens

- A cadeia é uma **subcadeia** de se aparece consecutivamente dentro de . Por exemplo, *cad* é uma subcadeia de *abracadabra*.
- Se temos a cadeia de comprimento e a cadeia de comprimento , a **concatenação** de e , é escrito Ao concatenar uma cadeia em si própria podemos usar notação de expoente.
- A **ordenação lexicográfica** de cadeias é a mesma que a ordenação familiar do dicionário, exceto que cadeias mais curtas precedem cadeias mais longas. Por exemplo, a ordenação lexicográfica de todas as cadeias sobre o alfabeto { 0,1 } é: (ε, 0, 1, 00, 01, 10, 11, 000, . . .).
- Uma **linguagem** é um conjunto de cadeias.

Cadeias e Linguagens

Exemplos:

- é uma cadeia, e é a subcadeia de .
- Se é uma cadeia, então é uma cadeia concatenada.
- e , então .

Cadeias e Linguagens

A **operação de concatenação** satisfaz às seguintes propriedades:

- ***Associativa***: .
- ***Elemento neutro***:

Em todos os casos os parênteses podem ser omitidos.

Cadeias e Linguagens

- Se representa um alfabeto, então denota o conjunto de todas as palavras possíveis sobre , e denota .
- Um **prefixo** (respectivamente, **sufixo**) de uma cadeia/palavra é qualquer sequência inicial (respectivamente, final) de símbolos dessa cadeia.

Cadeias e Linguagens

Exemplos:

- Se , então
- .

Lógica Booleana

- ***Lógica booleana*** é um sistema matemático construído em torno de dois valores: verdadeiro e falso.

- Os valores **verdadeiro** e **falso** são chamados os ***valores booleanos.***
- Podemos manipular valores booleanos com operações especialmente desenhadas, chamadas as ***operações booleanas.***
- As operações booleanas básicas são a **negação** , **conjunção** e **disjunção**

Lógica Booleana

Exemplos:

- $\neg 0 = 1$ e $\neg 1 = 0$
- $0 \wedge 0 = 0$, $0 \wedge 1 = 0$, $1 \wedge 0 = 0$ e $1 \wedge 1 = 1$
- $0 \vee 0 = 0$, $0 \vee 1 = 1$, $1 \vee 0 = 1$ e $1 \vee 1 = 1$

Lógica Booleana

- A operação de **ou exclusivo** ou **XOR** é designada por ⊕ (oplus). Se entre dois operadores um deles for igual a 1, então o resultado é 1.
- A operação de **igualdade**, escrita com o símbolo , é 1 se o seus dois operandos são iguais.
- A operação de **implicação**, escrita com o símbolo →, é 0 se seu primeiro operando é 1 e se seu segundo operando é 0; caso contrário é 1.

Lógica Booleana

Exemplos:

- $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$ e $1 \oplus 1 = 0$
- $00 = 1$, $01 = 0$, $10 = 0$ e $11 = 1$
- $0 \rightarrow 0 = 1$, $0 \rightarrow 1 = 1$, $1 \rightarrow 0 = 0$ e $1 \rightarrow 1 = 1$

Lógica Booleana

- A **lei distributiva** para E e OU permite manipulação de expressões booleanas, similar ao que é feito em adições e multiplicações.

Exemplos:

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

Lógica Booleana

- **Proposição** ou **sentença** é toda declaração que pode ser classificada como **Verdadeira** ou **falsa**.
- Tal declaração deve conter sujeito e predicado, ter apenas um entre os dois valores lógicos e ser declarativa (não ser exclamativa nem interrogativa).

Exemplos:

- Nove é diferente de cinco. ()
- Três é divisor de onze. (3|11)
- Dois é um número inteiro. ($2 \in \mathbb{Z}$)

Lógica Booleana

- **Sentenças abertas** são sentenças que contêm variáveis e cujo valor lógico (***Verdadeira*** ou ***falsa***) vai depender do valor atribuído à variável.
- Há duas maneiras de transformar sentenças abertas em proposições: (i) atribuir valor às variáveis; e (ii) utilizar quantificadores;
- O quantificador **universal** é indicado pelo símbolo ∀ e lido como "qualquer que seja", "para todo", "para cada".
- O quantificador **existencial** é indicado pelo símbolo ∃ e lido como "existe", "existe pelo menos um", "existe um".

Lógica Booleana

Exemplos:

- x + 1 = 7
- (∀x)(x +1 = 7), "qualquer que seja x, temos x +1 = 7"
- (∃x)(x +1 = 7), "existe um número x, tal que x +1 = 7"

Lógica Booleana

- Uma sentença quantificada com um quantificador universal, do tipo $(\forall x)(p(x))$, é **negada** substituindo-se o quantificador pelo **existencial** e negando-se p(x), ou seja, $(\exists x)(\neg p(x))$.
- Uma sentença quantificada com um quantificador existencial, do tipo $(\exists x)(p(x))$, é **negada** substituindo-se o quantificador pelo **universal** e negando-se p(x), ou seja, $(\forall x)(\neg p(x))$.

Exemplos:

- Sentença: ; negação:
- Sentença: ; negação:

DEFINIÇÕES, TEOREMAS E PROVAS

Definições, Teoremas e Provas

- ***Definições*** descrevem os objetos e noções que usamos. Precisão é essencial. É preciso deixar claro o que constitui o objeto e o que não constitui.
- Após termos definido vários objetos e noções, usualmente fazemos ***enunciados matemáticos*** sobre eles para expressar suas propriedades.

Exemplos:

- **Definição**: Um inteiro é considerado par se é divisível por 2.
- **Enunciado**: Para todo grafo G, a soma dos graus de todos os nós em G é um número par.

Definições, Teoremas e Provas

- Enunciados podem ser verdadeiros ou falsos. **Prova** é um argumento lógico de que o enunciado é verdadeiro em sentido absoluto.
- Um **teorema** é um enunciado matemático demonstrado como verdadeiro. São enunciados de interesse especial.
- Um **Lema** é um teorema que ajuda na prova de outro teorema.
- **Corolários** são teoremas ou sua prova rápida que ajudam a concluir facilmente que outros anunciados relacionados são verdadeiros.

Definições, Teoremas e Provas

Exemplos:

- **Teorema**: Se e são os comprimentos dos catetos de um triângulo retângulo e é o comprimento da hipotenusa, então .

Definições, Teoremas e Provas

- Uma **prova** é uma argumentação que mostra, de maneira indiscutível, que uma afirmação é verdadeira. É convincente em um sentido absoluto.
- A ´única maneira de determinar a veracidade ou a falsidade de um enunciado matemático é com uma prova matemática.
- Tipos mais frequentes de enunciados ocorrem na forma "P se e somente se Q" (P sse Q) ou PQ, que significa um enunciado de duas partes:

➢Direção de ida: "P somente se Q" ou P → Q

➢Direção reversa: "P se Q" ou P ← Q

Definições, Teoremas e Provas

- Um outro tipo de enunciado de múltiplas partes afirma que dois conjuntos A e B são iguais. Para provar que dois conjuntos são iguais, demonstre que todos os membros de um também é um membro do outro.
- Experimentar com exemplos é especialmente útil. Se o enunciado diz que todos os objetos de um certo tipo têm uma propriedade específica, escolha uns poucos objetos daquele tipo e observe que eles na realidade têm mesmo aquela propriedade.
- **Contra-exemplos** são usados para demonstrar que algum objeto rompe com as propriedades enunciadas como válidas.

Definições, Teoremas e Provas

Exemplo 1: Provar o enunciado: "para todo grafo G, a soma dos graus de todos os nós em G é um número par".

Definições, Teoremas e Provas

Exemplo 1: Provar o enunciado: "para todo grafo G, a soma dos graus de todos os nós em G é um número par".

Como provar??

Definições, Teoremas e Provas

- Uma prova bem-escrita é uma sequencia de enunciados.
- Escrever uma prova é importante, tanto para permitir que um leitor a entenda quanto para você ter certeza de que ela está livre de erros.

Definições, Teoremas e Provas

Dicas para se produzir uma prova:

- ***Seja paciente***. Encontrar provas leva tempo. Pesquisadores às vezes trabalham por semanas ou até anos para encontrar uma única prova.
- ***Volte a ela***. Dê uma olhada no enunciado que você quer provar, pense nele um pouco, então retorne uns poucos minutos ou horas mais tarde. Deixe a parte inconsciente, intuitiva de sua mente ter uma chance de trabalhar.

Definições, Teoremas e Provas

Dicas para se produzir uma prova:

- ***Seja claro.*** Quando você está construindo sua intuição para o enunciado que você está tentando provar, use figuras e/ou textos simples, claros. Além disso, quando você está escrevendo uma solução para uma outra pessoa ler, a clareza ajudará aquela pessoa a entendê-la.
- ***Seja conciso***. Brevidade ajuda a você a expressar ideias de alto nível sem se perder em detalhes.

Definições, Teoremas e Provas

Exemplo 1: Provar o enunciado: "para todo grafo G, a soma dos graus de todos os nós em G é um número par".

Toda aresta em G está conectada a dois nós. Cada aresta aumenta em uma unidade o grau de cada nó ao qual ela está conectada. Portanto, cada aresta contribui com 2 para a soma dos graus de todos os nós. Logo, se G tem e arestas, então a soma dos graus de todos os nós de G é 2e, que é um número par.

Definições, Teoremas e Provas

Exemplo 2: Provar uma das leis de DeMorgan: "para quaisquer dois conjuntos e , ”.

- Suponha que seja um elemento de . Então x não está em da definição do complemento de um conjunto, ou seja, não está em A nem em B. Em outras palavras, está em e em . Logo, a definição da intersecção de dois conjuntos mostra que está em .

Definições, Teoremas e Provas

Exemplo 2: Provar uma das leis de DeMorgan: "para quaisquer dois conjuntos e , ”.

- Para a outra direção, suponha que x esteja em . Então está em ambos e . Consequentemente, não está em nem em , e portanto não está na união de dois conjuntos. Logo, está no complemento da união desses dois conjuntos, isto é, está em , o que complementa a prova do teorema.

Definições, Teoremas e Provas

Tipos de provas:

1. **Prova por construção**
2. **Prova por contradição**
3. **Prova por indução**

Definições, Teoremas e Provas

Tipos de provas:

1. **Prova por construção:** Para teoremas que enunciam que um objeto existe, demonstramos como construir esse objeto.
2. **Prova por contradição:** Assumimos que o teorema é falso e mostramos que essa suposição leva a uma consequência falsa.
3. **Prova por indução:** Demonstra que o teorema é correto em todos os passos ou para todas as entradas.

Tipos de Provas: Prova por construção

Exemplo 3: Para cada número par maior que 2, existe um grafo 3-regular com nós.

Tipos de Provas: Prova por construção

Exemplo 3: Para cada número par maior que 2, existe um grafo 3-regular com nós.

- Seja . Construa o grafo , com nós. Seja e . Desenhe os nós desse grafo escritos consecutivamente ao redor da circunferência de um círculo. Nesse caso, as arestas descritas na linha superior de E ligam pares adjacentes ao longo do círculo. As arestas descritas na linha inferior de E ligam nós em lados opostos do círculo. Dessa forma fica demonstrado que todo nó em G tem grau 3.

Tipos de Provas: Prova por contradição

Jack vê Jill, que acaba de chegar da rua. Observando que ela está completamente enxuta, ele sabe que não está chovendo. Sua “prova” de que não está chovendo é que, se estivesse chovendo (a suposição de que o enunciado é falso), Jill estaria molhada (a consequência obviamente falsa). Portanto não pode estar chovendo.

Tipos de Provas: Prova por contradição

Exemplo 4: é irracional.

Tipos de Provas: Prova por contradição

Exemplo 4: é irracional.

(Parte 1): Primeiramente, vamos supor que é racional. Assim, , onde e são inteiros.

Se e são divisíveis pelo mesmo inteiro maior que 1, divida ambos por esse inteiro. Fazer isso não muda o valor da fração.

Tipos de Provas: Prova por contradição

Exemplo 4: é irracional.

(Parte 2): Agora, pelo menos um, dentre e , não é par. Multiplicamos ambos os lados por e obtemos . Elevando ambos os lados ao quadrado temos . Em virtude de ser 2 vezes o inteiro , sabemos que é par. Assim, também é par, pois o quadrado de um número ímpar é sempre ímpar. Portanto, podemos escrever para algum inteiro . Então, substituindo por , obtemos . Dividindo ambos os lados por 2 obtemos . Mas esse resultado mostra que é par e, assim, é par. Dessa forma, estabelecemos que tanto quanto são pares. Mas tínhamos reduzido e de modo que não fossem pares, o que é uma contradição.

Tipos de Provas: Prova por indução

- Na linguagem usual, indução se refere à extração de conclusões gerais ao se examinar vários fatos particulares.
- **Prova por indução** é um método avançado usado para mostrar que todos os elementos de um conjunto infinito têm uma propriedade especificada.
- Usamos rotineiramente para mostrar que um programa fur odas as

Indução matemática

Seja $\mathbb{N} = \{1, 2, 3, ...\}$ e suponha uma propriedade P, tal que P possa ser aplicada a algum elemento em $\mathbb{N}$.
O objetivo é verificar que, $\forall k \in \mathbb{N}$, $P(k)$ é verdadeiro.

Tipos de Provas: Prova por indução

- Toda prova por indução consiste de duas partes, o ***passo da indução*** e a ***base***.
- Prova por indução = Base de indução (B.I) + passo de indução (P.I.)
- Passo de indução começa com uma hipótese de indução (H.I.)
- Base de indução: P(1) é verdadeira
- Passo de indução:
  - *Hipótese de indução*: Supor que é verdadeira, .
  - *Passo*: Se é verdadeiro, provar que é verdadeiro.

Tipos de Provas: Prova por indução

Exemplo 5: Provar por indução a correção da fórmula usada para calcular as prestações mensais da casa própria. Ao comprar uma casa, muitas pessoas tomam algum dinheiro emprestado (financiamento) e o pagam durante alguns anos. Uma quantidade fixa é paga a cada mês para cobrir os juros, assim como uma parte do montante original.

Tipos de Provas: Prova por indução

Exemplo 5: Provar por indução a correção da fórmula usada para calcular as prestações mensais da casa própria. Ao comprar uma casa, muitas pessoas tomam algum dinheiro emprestado (financiamento) e o pagam durante alguns anos. Uma quantidade fixa é paga a cada mês para cobrir os juros, assim como uma parte do montante original.

Passos da solução

- Nomes e significados das variáveis
- Acontecimento mês a mês
- Prova

Tipos de Provas: Prova por indução

Passos da solução

- *Nomes e significados das variáveis*:
  - P: principal (montante do empréstimo)
  - I: taxa de juros anual (I = 6% = 0.06)
  - Y : pagamento mensal
  - M: multiplicador mensal (taxa de mudança do valor mensal) sendo
- *Acontecimento mês a mês*:

(a) Montante do empréstimo (P) cresce devido ao multiplicador mensal (M)

(b) Montante (P) tende a diminuir devido ao pagamento mensal (Y)

- *Prova*

Tipos de Provas: Prova por indução

Seja o montante restante do empréstimo após o mês. Então:

- montante do empréstimo original.
- montante do empréstimo após **1** mês.
- montante do empréstimo após **2** meses.

- Vamos provar um teorema por indução sobre que dá uma fórmula para .

Tipos de Provas: Prova por indução

Fórmula:

Base da indução

Caso base: Provar que a fórmula é verdadeira para

Com isso, obtemos:

é verificável, portanto a indução é **verdadeira.**

Tipos de Provas: Prova por indução

Passo da indução:

Hipótese da indução: Para , suponha que a fórmula abaixo seja verdadeira

Provar que a fórmula é verdadeira para

Tipos de Provas: Prova por indução

Da definição de a partir de , sabemos que:

Usando a hipótese de indução para calcular

Tipos de Provas: Prova por indução

Distribuindo e reescrevendo temos:

Logo,

Portanto, a fórmula é verdadeira para , o que prova o teorema.

Tipos de Provas: Prova por indução

Exercícios:

1. Use a equação anterior para derivar uma fórmula para calcular o tamanho do pagamento mensal para uma amortização em termos do principal , taxa de juros I, e o número de pagamentos . Assuma que, após pagamentos tiverem sido feitos, o montante do empréstimo é reduzido a 0. Use a fórmula para calcular o montante em dólares de cada pagamento mensal para uma amortização de 30-anos com 360 pagamentos mensais sobre um montante de empréstimo inicial de $100.000 com uma taxa anual de juros de 5%.

Tipos de Provas: Prova por indução

Exercícios:

2. Encontre o erro na seguinte prova de que 2 = 1.

Considere a equação . Multiplique ambos os lados por para obter . Subtraia de ambos os lados para obter . Agora fatore cada lado, , e divida cada lado por , para chegar em . Finalmente, faça e b iguais a 1, o que mostra que 2 = 1.