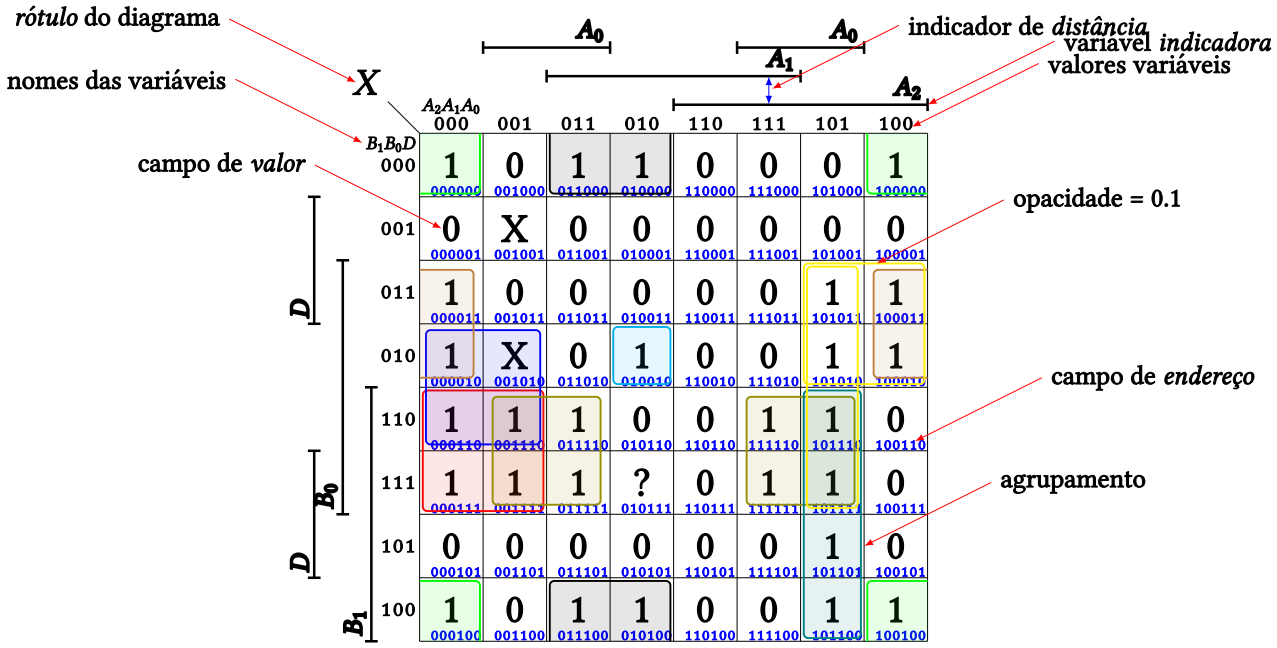
Fundamentos de Matemática para Computação



$$X = \overline{A_2}\,\overline{A_1}\,B_1\,B_0 + \overline{A_2}\,\overline{A_1}\,B_0\,\overline{D} + \overline{A_1}\,\overline{A_0}\,\overline{B_0}\,\overline{D} + \overline{A_2}\,A_1\,\overline{B_0}\,\overline{D} + A_0\,B_1\,B_0 + \overline{A_1}\,\overline{A_0}\,\overline{B_1}\,B_0 + A_2\,\overline{A_1}\,A_0\,B_0 + \overline{A_2}\,A_1\,\overline{A_0}\,\overline{B_1}\,B_0\,\overline{D}$$

ou

$$X = \overline{A_2}\,\overline{A_1}\,B_1\,B_0 + \overline{A_2}\,\overline{A_1}\,B_0\,\overline{D} + \overline{A_1}\,\overline{A_0}\,\overline{B_0}\,\overline{D} + \overline{A_2}\,A_1\,\overline{B_0}\,\overline{D} + A_0\,B_1\,B_0 + \overline{A_1}\,\overline{A_0}\,\overline{B_1}\,B_0 + A_2\,\overline{A_1}\,\overline{B_1}\,B_0 + \overline{A_2}\,A_1\,\overline{A_0}\,\overline{B_1}\,B_0\,\overline{D}$$

Notas de Aula para Sistemas de Informação para Internet — Versão 0.2

Conteúdo

Capítulo 1

Pensamento Lógico

O negócio dos matemáticos é afirmar coisas precisamente. Quando você lê uma sentença matemática, deve levar a sério cada palavra; uma boa linguagem matemática transmite uma mensagem clara, sem ambiguidades. Para ler e escrever matemática, você deve praticar a arte do pensamento lógico. O objetivo deste capítulo é ajudá-lo a se comunicar matematicamente pelo entendimento básico da lógica. Atenção: lógica matemática pode ser difícil — especialmente se for a primeira vez que você vê isso. Este capítulo começa com o estudo da lógica formal, ou simbólica, e depois aplica esse estudo à linguagem matemática. Espere que as coisas sejam um pouco nebulosas no começo, mas no final (esperamos) a neblina irá clarear.

1.1 Lógica Formal

Notação é uma parte importante da linguagem matemática. Os quadros-negros dos matemáticos geralmente estão cheios de todo tipo de caracteres e símbolos estranhos; tal exibição pode ser intimidadora para os novatos, mas existe uma boa razão para comunicar dessa maneira. Geralmente o ato de reduzir um problema a uma linguagem simbólica nos ajuda ver o que realmente está acontecendo. Em vez de operar no mundo vago da prosa, traduzimos um problema para notação matemática e depois realizamos manipulações simbólicas bem definidas sobre essa notação. Essa é a essência de uma poderosa ferramenta chamada *formalismo*. Nesta seção, exploramos como uma abordagem formal à lógica pode ajudar a evitar erros de raciocínio.

Nota:

Usaremos a palavra *formal* para descrever o processo que consiste em manipular notações. Geralmente as pessoas usam essa palavra para significar "rigoroso", mas essa não é a nossa intenção. Um argumento formal pode ser rigoroso, mas um argumento que não depende de símbolos também pode.

Uma característica agradável do formalismo é permitir que você trabalhe sem ter que pensar sobre o que os símbolos significam. Nesse sentido, lógica formal é na verdade um "não pensamento lógico". Por que isso é uma vantagem? Cálculos formais são menos propensos a erros. Você já está familiarizado com esses fenômenos: muito da aritmética que você aprendeu na escola era formal. Você tem um algoritmo simbólico bem definido para multiplicar números usando lápis e papel, e consegue multiplicar de forma bem eficiente números de três dígitos sem nem pensar muito sobre o que realmente está fazendo. Claro, o formalismo não faz sentido se você não sabe o que está fazendo; no final de qualquer cálculo formal, é importante ser capaz de interpretar os resultados.

1.2 Conectivos e Proposições

A fim de formalizar a lógica, precisamos de um sistema para traduzir afirmações em símbolos. Vamos começar com uma definição precisa de *sentença*.

Definição 1.1

Uma *sentença* (também conhecida por *proposição*) é uma frase declarativa que pode ser falsa ou verdadeira, mas não as duas ao mesmo tempo.

São exemplos de sentenças:

☞ 7 é ímpar.

☞ 1 + 1 = 4

☞ Se está chovendo, então o chão está molhado.

☞ O nosso professor é de Marte.

Nome	Símbolo
e (conjunção)	$\wedge$
ou (disjunção)	$\vee$
não (negação)	$\neg$
implica (se... então)	$\rightarrow$
se, e somente se	$\leftrightarrow$

Tabela 1.1: Os cinco conectivos lógicos.

Note que não precisamos ser capazes de decidir se a sentença é verdadeira ou falsa para que seja uma sentença. Ou nosso professor é de Marte, ou nosso professor não é de Marte, ainda que não estejamos certos de qual é o caso.

Como pode uma frase declarativa falhar em ser uma sentença? Existem duas maneiras principais. Uma frase declarativa pode conter um termo não especificado:

x é par.

Nesse caso, x é chamado de *variável livre*. A veracidade da frase depende do valor de x, logo, se esse valor não é especificado, não podemos considerar que essa frase é uma sentença. Um segundo tipo de frase declarativa que não é uma sentença ocorre quando uma frase é *autorreferencial*:

Esta frase é falsa.

Não podemos decidir se essa frase é verdadeira ou não. Se dizemos que é verdadeira, então ela diz ser falsa; se dizemos que é falsa, então ela parece ser verdadeira.

Geralmente, uma sentença complicada consiste em várias sentenças simples unidas por palavras como "e", "ou", "se... então" etc. Essas palavras conectivas são representadas pelos *cinco conectivos lógicos* mostrados na Tabela 1.1. Conectivos lógicos são úteis para decompor sentenças compostas em sentenças mais simples. Eles ressaltam propriedades lógicas importantes da sentença.

A fim de utilizar um sistema formal para a lógica, devemos ser capazes de *traduzir* uma sentença em português em sua contrapartida formal. Fazemos isso atribuindo letras para sentenças simples e depois construindo expressões com conectivos.

Exemplo 1.1

Se p é a sentença "você está usando sapatos" e q é a sentença "você não pode cortar as unhas do pé", então

$$p \rightarrow q$$

representa a sentença: "Se você está usando sapatos, então não pode cortar as unhas do pé." Podemos optar por expressar essa sentença de formas diferentes em português: "Você não pode cortar as unhas do pé se está usando sapatos", ou "Usando sapatos é impossível cortar as unhas do pé". A sentença $\neg q$ é traduzida literalmente em "Não é o caso de você não poder cortar as unhas do pé". E claro, em português, preferiríamos dizer simplesmente, "Você pode cortar as unhas do pé", mas isso envolve usar lógica, como veremos na seção seguinte.

1.2.1 Tabelas Verdade

Ainda não terminamos de configurar o nosso sistema formal de lógica porque não fomos específicos a respeito dos significados dos conectivos de lógica. E claro que os nomes de cada conectivo sugerem como eles devem ser usados, mas, a fim de elaborar sentenças matematicamente precisas, precisamos saber exatamente o que cada conectivo significa.

Definir o significado de um símbolo matemático é mais difícil do que pode parecer. Até mesmo o símbolo + de aritmética ordinária é problemático. Embora todos nós tenhamos um entendimento intuitivo de adição — ela descreve como combinar duas quantidades —, é difícil expressar esse conceito em palavras sem ter que apelar para a nossa intuição. O que "combinar" significa, exatamente? O que são "quantidades", na verdade? Um jeito simples, mas obviamente não prático, de definir o sinal de + seria listar todos os problemas possíveis de adição, como na Tabela 1.2. E claro que uma tabela como essa não teria fim, mas nos daria, em teoria, uma definição precisa do sinal de +.

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	2
2	1	3
⋮	⋮	⋮

Tabela 1.2: Definir o sinal de + listando todos os problemas possíveis de adição requereria uma tabela infinita.

É mais fácil lidar com a situação em lógica. Qualquer sentença tem dois valores possíveis: verdadeiro (V) ou falso (F). Então, quando usamos variáveis como p ou q para uma sentença lógica, podemos considerá-los incógnitas que podem ter um dos dois valores: V ou F. Isso torna possível definir o significado de cada conectivo usando tabelas; em vez de termos infinitos possíveis valores para os números x e y, temos somente duas escolhas para cada variável p e q.

Agora, vamos estipular o significado de cada conectivo lógico listando os valores V/F para cada caso possível. O conectivo "não" é o exemplo mais simples, — Se p é verdadeira, então $\neg p$ deve ser falsa, e vice-versa.

p	$\neg p$
V	F
F	V

Essa tabela de valores é chamada de *tabela verdade*; ela define os valores V/F para os conectivos. Os conectivos "e", "ou", "se, e somente se" e "implica" são definidos pelas seguintes tabelas verdades. Uma vez que temos duas variáveis, e cada uma pode ser tanto V como F, precisamos de quatro casos que são distribuídos nas quatro linhas de cada tabela.

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

p	q	$p \leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

A definição do conectivo "e", indicado pelo sinal $\wedge$, é o que você deveria esperar: para que $p \wedge q$ sejam verdade, p deve ser verdade e q deve ser verdade. O conectivo "ou", indicado pelo sinal $\vee$ é um pouco menos óbvio. Note que a nossa definição estipula que $p \vee q$ é verdade sempre que p for verdade, ou q for verdade, ou ambos forem verdade. Isso pode ser diferente da forma como o "ou" é utilizado na linguagem cotidiana. Quando lhe oferecem "sopa ou salada" em um restaurante, o garçom não espera que você peça por "ambos".

O conectivo "se e somente se" diz que duas sentenças têm exatamente os mesmos valores V /F. Dessa forma, $p \leftrightarrow q$ é verdade quando p e q coincidem em valor e será falsa caso isso não ocorra.

O conectivo "implica" tem a definição menos intuitiva. Para entendermos a motivação dessa definição, relembre o Exemplo 1.1 . A fim de demonstrar que a sentença $p \rightarrow q \equiv$[1] "Se você está usando sapatos, então não pode cortar as unhas do pé." é falsa, você deveria ser capaz de cortar as unhas do pé enquanto usasse sapatos. Em qualquer outra situação, você teria que admitir que a sentença não é falsa (e, se uma sentença não é falsa, ela deve ser verdadeira). Se você não está usando sapatos, então talvez consiga cortar as unhas do pé ou talvez não consiga por alguma outra razão. Isso não contradiz a sentença $p \rightarrow q$.

Dito de outra maneira, se você vive em um mundo sem sapatos, então a sentença é verdadeira por vacuidade: uma vez que você não pode nunca, de fato, usar sapatos, não é falso dizer que "Se você está

[1] O símbolo ≡ representa equivalência em Lógica Formal. Ele é usado como um caso especial de igualdade.

usando sapatos," então tudo é possível. Isso explica as duas últimas linhas da tabela verdade; se p é falsa, então $p \to q$ é verdade, não importa o que seja q.

1.3 Equivalências Lógicas

Definição 1.2

Duas sentenças são *logicamente equivalentes* se têm os mesmos valores V/F para todos os casos, ou seja, se elas têm as mesmas tabelas verdades.

Existem algumas equivalências lógicas que aparecem frequentemente em matemática, e também na vida em geral.

Exemplo 1.2

Considere o seguinte teorema de geometria do Ensino Médio:

Se um quadrilátero tem um par de lados paralelos, então ele tem um par de ângulos suplementares (ângulos cuja soma é 180°).

Esse teorema é da forma $p \to q$, em que p é a sentença de que o quadrilátero tem um par de lados paralelos, e q é a sentença de que o quadrilátero tem um par de ângulos suplementares.
Podemos ainda afirmar um teorema diferente, representado por $\neg q \to \neg p$:

Se o quadrilátero não tem um par de ângulos suplementares, então ele não tem um par de lados paralelos.
Sabemos que esse segundo teorema é logicamente equivalente ao primeiro porque a sentença formal $p \to q$ é logicamente equivalente à sentença formal $\neg q \to \neg p$, como mostra a seguinte tabela verdade:

p	q	$p \to q$	$\neg q$	$\neg p$	$\neg q \to \neg p$
V	V	V	F	F	V
V	F	F	V	F	F
F	V	V	F	V	V
F	F	V	V	V	V

Note que a coluna $p \to q$ é igual à coluna $\neg q \to \neg p$. Uma vez que o primeiro teorema é um teorema de geometria verdadeiro, concluímos que o segundo também é.
Agora considere a seguinte variação para esse teorema:

Se um quadrilátero tem um par de ângulos suplementares, então ele tem um par de lados paralelos.
Essa sentença é da forma $q \to p$. Mas a tabela verdade a seguir mostra que $q \to p$ não é logicamente equivalente a $p \to q$, porque os valores V/F são diferentes na segunda e terceira linhas.

q	p	$q \to p$
V	V	V
V	F	V
F	V	F
F	F	V

De fato, essa última sentença geralmente não é verdadeira em geometria. (Você consegue desenhar um exemplo de um quadrilátero para o qual essa sentença não seja verdadeira?)

A sentença $\neg q \to \neg p$ é chamada de **contrapositiva** de $p \to q$, e a sentença $q \to p$ é chamada de **recíproca**. A tabela verdade anterior nos prova que, para qualquer sentença s, a contrapositiva de s é logicamente equivalente a s, enquanto a recíproca de s pode não ser logicamente equivalente.

Existem muitas situações em que assumir a recíproca pode causar problemas. Por exemplo, suponha que a seguinte sentença seja verdadeira:

Se uma empresa não participa de práticas ilegais de contabilidade, então uma auditoria não encontrará evidências de irregularidades.

É certamente razoável assumir isso: não pode haver evidências de irregularidades uma vez que não existem irregularidades. No entanto, é provável que a recíproca não seja verdadeira:

Se uma auditoria não encontra nenhuma evidência de irregularidades, então a empresa não participa de práticas ilegais de contabilidade.

Afinal de contas, é possível que os auditores tenham cometido erros.

Nesse momento, você poderia alegar que o uso de lógica formal parece muito complicado para apenas verificar deduções como esse último exemplo. Esse tipo de coisa é apenas senso comum, certo? Bem, talvez. Mas algo que parece óbvio para você pode não ser óbvio para outra pessoa. Além disso, nosso sistema de lógica formal irá lidar com situações mais complicadas, em que o nosso senso comum costuma falhar. A solução para o próximo exemplo usa lógica formal. Antes que você veja a solução, tente resolver o problema usando o "senso comum". Embora a abordagem formal leve um pouco mais de tempo, ela resolve qualquer dúvida que você tenha sobre o seu próprio processo de raciocínio.

Exemplo 1.3

Se Alcides está atrasado, então Belmiro está atrasado, e, se Alcides e Belmiro estão ambos atrasados, então a aula é chata. Suponha que a aula não seja chata. O que você pode concluir a respeito de Alcides?

Solução: Vamos começar traduzindo a primeira frase em símbolos de lógica, usando as seguintes sentenças:

p = Alcides está atrasado.
q = Belmiro está atrasado.
r = A aula é chata.

Seja S a sentença "Se Alcides está atrasado, então Belmiro está atrasado, e, se Alcides e Belmiro estão ambos atrasados, então a aula é chata." Em símbolos, S é traduzida como:

$$S = (p \rightarrow q) \wedge [(p \wedge q) \rightarrow r]$$

Agora vamos construir uma tabela verdade para S. Fazemos isso construindo tabelas verdade para as diferentes partes de S, começando com as de dentro dos parênteses e resolvendo do jeito que já conhecemos.

#	p	q	r	$p \rightarrow q$	$p \wedge q$	$(p \wedge q) \rightarrow r$	S
1.	V	V	V	V	V	V	V
2.	V	V	F	V	V	F	F
3.	V	F	V	F	F	V	F
4.	V	F	F	F	F	V	F
5.	F	V	V	V	F	V	V
6.	F	V	F	V	F	V	V
7.	F	F	V	V	F	V	V
8.	F	F	F	V	F	V	V

Verifique que a última coluna é o resultado de "e-zar" a coluna de $p \rightarrow q$ com a coluna $(p \wedge q) \rightarrow r$. Estamos interessados nos possíveis valores de p. Sabemos que S é verdadeira, logo podemos eliminar as linhas 2, 3 e 4, as linhas em que S é falsa. Se também assumirmos que a aula não é chata, então poderemos eliminar as linhas em que r é verdadeira, isto é, as linhas numeradas com números ímpares. As linhas que sobraram são as únicas com possíveis valores V/F para p, q, e r, ou seja, as linhas 6 e 8. Em ambas as linhas, p é falsa. Em outras palavras, Alcides não está atrasado.

1.3.1 Exercícios propostos

1 Sejam dadas as seguintes sentenças:
p = "Tem água nos cilindros."
q = "A junta do cabeçote está vazando."
r = "O carro vai pegar."

a) Traduza a sentença seguinte para símbolos de lógica formal.

"Se a junta do cabeçote está vazando e tem água no cilindro, então o carro não vai pegar."

b) Traduza a seguinte sentença formal para o português comum:

$$r \rightarrow \neg(q \vee p)$$

2 Seja s a sentença seguinte:
"Se você está estudando muito, então está ficando acordado até tarde da noite."

a) Dê a recíproca de s.

b) Dê a contrapositiva de s.

3 Dizemos que dois pares ordenados (a, b) e (c, d) são iguais quando $a = c$ e $b = d$. Seja s a sentença seguinte.

Se $(a, b) = (c, d)$, então $a = c$.

a) Esta sentença é verdadeira?

b) Escreva a recíproca de s.

c) A recíproca de s é verdadeira? Explique.

4 Use a tabela verdade para provar que a sentença

$$[(p \vee q) \wedge (\neg p)] \rightarrow q$$

é sempre verdadeira, independentemente do que sejam p e q.

5 Sejam dadas as seguintes sentenças:
p = "Amauri está com fome."
q = "A geladeira está vazia."
r = "Amauri está zangado."

a) Use os conectivos para traduzir a sentença seguinte para a lógica formal:

Se Amauri está com fome e a geladeira está vazia, então Amauri está zangado.

b) Construa a tabela verdade para a sentença em a).

c) Suponha que a sentença dada em a) seja verdadeira, e suponha também que Amauri não esteja zangado e a geladeira esteja vazia. Amauri está com fome? Justifique sua resposta usando a tabela verdade.

6 Geralmente podemos simplificar uma sentença complicada em lógica formal. Por exemplo, considere a sentença $S = (p \wedge q) \vee (p \wedge \neg q)$.

a) Construa a tabela verdade para S.

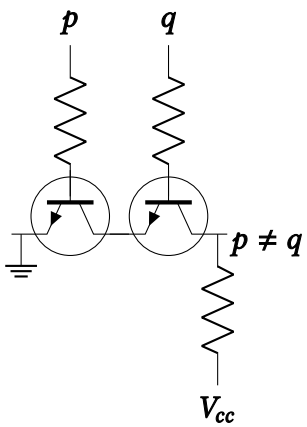
b) Ache uma expressão simplificada que seja logicamente equivalente a S.

7 O conectivo NAND é simbolizado por $\uparrow$ e definido pela seguinte tabela verdade ao lado. Use tabelas verdade para mostrar que $p \uparrow q$ é logicamente equivalente a $\neg(p \wedge q)$.

p	q	$p \uparrow q$
V	V	F
V	F	V
F	V	V
F	F	V

(Isso explica o nome NAND = not AND, "não e" em português.)

8 O conectivo NAND é importante porque facilmente se constrói um circuito eletrônico que calcula o NAND de dois sinais (veja o esquema ao lado). Esse circuito é chamado de *porta lógica*. Além do mais, é possível construir portas lógicas para outros conectivos lógicos usando apenas portas NAND. Prove esse fato mostrando, através de tabelas verdade, as equivalências seguintes:



a) $(p \uparrow q) \uparrow (p \uparrow q)$ é logicamente equivalente a $p \wedge q$.

b) $(p \uparrow p) \uparrow (q \uparrow q)$ é logicamente equivalente a $p \vee q$.

c) $p \uparrow (q \uparrow q)$ é logicamente equivalente a $p \rightarrow q$.

1.4 Lógica Proposicional

Depois de ter trabalhado nos exercícios da seção anterior, você deve ter notado uma séria limitação no uso de tabelas verdade. Cada vez que você adiciona uma sentença na tabela verdade, você deve dobrar o número de linhas. Isso torna a análise de tabelas verdade trabalhosa para todos os exemplos que não aqueles mais simples.

Nesta seção iremos desenvolver um sistema de regras para manipular fórmulas em lógica simbólica. Esse sistema, chamado de *cálculo proposicional*, irá nos permitir fazer deduções lógicas formalmente. Existem pelo menos três razões para que façamos isso:

a) Esses métodos formais são úteis para analisar problemas complexos de lógica, especialmente quando o uso de tabelas verdade é pouco prático.

b) Esses métodos formais são úteis para analisar As regras de dedução que iremos estudar são comumente usadas na discussão matemática.

c) O sistema de regras de dedução e sequências de demonstração é um exemplo simples de demonstração matemática.

Dessas três, a última é a mais importante. O processo de escrever sequências de demonstração em cálculo proposicional serve, dentre outras coisas, para análise de algoritmos em outras áreas da matemática.

Existem algumas sentenças em lógica formal que são sempre verdadeiras, não importam quais sejam os valores V/F das sentenças componentes. Por exemplo, a tabela verdade para $(p \wedge q) \to p$ é a seguinte:

p	q	$p \wedge q$	$(p \wedge q) \to p$
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	V

Uma sentença como essa é chamada de **tautologia**, e escrevemos

$$(p \wedge q) \Rightarrow p$$

para indicar esse fato. A notação $A \Rightarrow B$ significa que a sentença $A \Rightarrow B$ é verdadeira em todos os casos; em outras palavras, a tabela verdade para $A \Rightarrow B$ é composta apenas por Vs. Analogamente, o símbolo $\Leftrightarrow$ denota a tautologia contendo o conectivo $\leftrightarrow$.

Quando a tautologia é da forma $(C \wedge D) \Rightarrow E$, é comum escrever $\left.\begin{matrix} C \\ D \end{matrix}\right\} \Rightarrow E$ como alternativa. Essa notação destaca o fato de que, se você conhece tanto C quanto D, então você pode concluir E. O uso do conectivo $\wedge$ é implícito.

Exemplo 1.4

Use a tabela verdade para provar $\left.\begin{matrix} p \\ p \to q \end{matrix}\right\} \Rightarrow q$.

Solução: Seja S a sentença $[p \wedge (p \to q)] \to q$. Vamos construir a nossa tabela verdade determinando as partes de S e trabalhando de dentro para fora dos parênteses. Uma vez que a coluna para S é toda composta de Vs, fica provado que S é uma tautologia.

p	q	$p \to q$	$p \wedge (p \to q)$	S
V	V	V	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	V

A tautologia do Exemplo 1.4 é conhecida como *modus ponens*, o que em latim significa "modo afirmativo". Esse conceito remonta aos filósofos estoicos da Grécia antiga, que afirmavam da seguinte forma:

Se o primeiro, então o segundo;
mas o primeiro;
portanto o segundo.

Um resultado relacionado usando negação é chamado *modus tollens* ("modo de negação"). Nos símbolos de lógica, essa tautologia se escreve da seguinte forma (perceba como é usada a contrapositiva):

$$\left.\begin{array}{l}\neg q \\ p \to q\end{array}\right\} \Rightarrow \neg p$$

Também existem as sentenças em lógica formal que nunca são verdadeiras. Uma sentença cuja tabela verdade é composta apenas por Fs é chamada de *contradição*.

Exemplo 1.5

Use uma tabela verdade para mostrar que $p \wedge \neg p$ é uma contradição.

Solução:

p	$\neg p$	$p \wedge \neg p$
V	F	F
F	V	F

Em outras palavras, uma sentença e sua negação nunca podem ser ambas verdadeiras.

Uma sentença em lógica formal que não é uma tautologia nem uma contradição é chamada de *contingência*. Uma contingência tem tanto Vs quanto Fs em sua tabela verdade, assim sua verdade é "contingente" nos valores V/F de suas sentenças componentes. Por exemplo, $p \wedge q$, $p \vee q$ e $p \to q$ são todas contingências.

1.5 Regras de Dedução

Tautologias são importantes porque mostram como uma sentença pode ser logicamente deduzida de outra. Por exemplo, suponha que saibamos que as sentenças seguintes são verdadeiras:

Nosso professor não é dono de uma espaçonave.
Se o nosso professor é de Marte, então nosso professor é dono de uma espaçonave.

Podemos aplicar a tautologia *modus tollens* para deduzir que "Nosso professor não é de Marte". Esse é um argumento válido, ou dedução, que nos permite concluir essa última sentença dadas as duas primeiras.

Toda tautologia pode ser usada como uma regra que justifica a dedução de uma nova sentença a partir de uma antiga. Existem dois tipos de regras de dedução: *regras de equivalência* e *regras de inferência*.

Regras de equivalência descrevem equivalências lógicas, enquanto regras de inferência descrevem quando uma sentença mais fraca pode ser deduzida de uma sentença mais forte. As regras de equivalência dadas na Tabela 1.3 podem ser verificadas usando tabelas verdade, onde representamos qualquer contradição por F e qualquer tautologia por V. Se A e B são sentenças (possivelmente compostas de muitas outras sentenças unidas por conectivos), então a tautologia $A \Leftrightarrow B$ é uma outra forma de dizer que A e B são logicamente equivalentes.

Uma regra de equivalência da forma $A \Leftrightarrow B$ pode fazer três coisas:

① Dado A, deduz-se B.

② Dado B, deduz-se A.

Equivalência	Nome
$p \Leftrightarrow \neg\neg p$	dupla negação
$p \to q \Leftrightarrow \neg p \vee q$	implicação
$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$	leis de De Morgan
$p \vee q \Leftrightarrow q \vee p$ $p \wedge q \Leftrightarrow q \wedge p$	comutatividade
$p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$ $p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$	associatividade
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	distributividade
$p \vee F \Leftrightarrow p$ $p \wedge V \Leftrightarrow p$	propriedade da identidade

Tabela 1.3: Regras de Equivalência

③ Dada uma sentença contendo a sentença A, deduz-se a mesma sentença, mas com a sentença A substituída pela sentença B.

A terceira opção é uma forma de substituição. Por exemplo, dada a seguinte sentença:

Se Melinda não está doente e Melinda não está cansada, então Melinda pode brincar.

podemos deduzir o seguinte usando as leis de De Morgan:

Se não for o caso de Melinda estar doente ou cansada, então Melinda pode brincar.

Além das regras de equivalência, existem também regras de inferência para lógica proposicional. Diferentemente das regras de equivalência, as regras de inferência funcionam somente em uma direção. Uma regra de inferência da forma $A \Rightarrow B$ permite que se faça somente uma coisa:

① Dado A, deduza B.

Em outras palavras, você pode concluir uma sentença mais fraca, B, se já estabeleceu uma sentença mais forte, A. Por exemplo, *modus tollens* é uma regra de inferência: a sentença mais fraca

B = "Nosso professor não é de Marte".

resulta de uma sentença mais forte

A = "Nosso professor não é dono de uma espaçonave, e se o nosso professor é de Marte, então ele é dono de uma espaçonave."

Se A é verdadeira, então B deve ser verdadeira, mas não vice-versa. (Nosso professor pode ser dono de uma espaçonave e ser de Júpiter, por exemplo.) A Tabela 1.4 lista algumas regras úteis de inferência, todas as quais podem ser verificadas usando tabelas verdade.

Inferência	Nome
$\left.\begin{matrix} p \\ q \end{matrix}\right\} \Rightarrow p \wedge q$	conjunção
R $\left.\begin{matrix} p \\ p \rightarrow q \end{matrix}\right\} \Rightarrow q$	*modus ponens*
$\left.\begin{matrix} \neg q \\ p \rightarrow q \end{matrix}\right\} \Rightarrow \neg p$	*modus tollens*
$p \wedge q \Rightarrow p$	simplificação
$p \Rightarrow p \vee q$	adição

Tabela 1.4: Regras de Inferência

1.5.1 Sequências de Prova

Agora temos ferramentas suficientes para deduzir novas tautologias das já conhecidas. Uma *sequência de prova* (ou *sequência de demonstração*) é uma sequência de sentenças e razões para justificar uma asserção da forma $A \implies C$. A primeira sentença, A, é dada. A sequência de prova pode então listar sentenças B_1, B_2, B_3, etc., desde que cada nova sentença possa ser deduzida de uma ou mais sentenças prévias usando alguma regra de dedução. E claro que essa sequência de sentenças deve culminar em C, a sentença que estamos tentando provar a partir de A.

Exemplo 1.6

Escreva uma sequência de prova para a asserção $\left.\begin{matrix} p \\ p \rightarrow q \\ q \rightarrow r \end{matrix}\right\} \Rightarrow r$.

Solução:

Sentença	Razões
1. p	dada
2. $p \rightarrow q$	dada
3. $q \rightarrow r$	dada
4. q	*modus ponens*, 1, 2
5. r	*modus ponens*, 4, 3

Toda vez que provamos alguma coisa, temos uma nova regra de inferência. As regras na Tabela 1.4 são suficientes para começarmos, mas deveríamos nos sentir livres para usar as asserções já provadas

em provas futuras. Por exemplo, a asserção provada no Exemplo 1.6 ilustra a propriedade transitiva do conectivo $\rightarrow$.

Uma outra coisa a ser notada no Exemplo 1.6 é que ele foi bastante fácil — só tivemos que aplicar o *modus ponens* duas vezes. Compare isso com a abordagem da tabela verdade: a tabela verdade para $[p \wedge (p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow r$ consistiria em oito linhas e várias colunas. Tabelas verdade são mais fáceis de ser feitas, mas também podem ser muito mais tediosas.

Sequências de prova devem lembrar os tipos de prova que você costumava fazer na geometria no ensino médio. As regras são simples: comece com o que é dado, veja o que você consegue deduzir, finalize com o que você está tentando provar. Aqui está um exemplo mais difícil:

Exemplo 1.7

Prove $\left.\begin{array}{l} p \vee q \\ \neg p \end{array}\right\} \Rightarrow q$.

Solução:

Sentença	Razões
1. $p \vee q$	dada
2. $\neg p$	dada
3. $\neg(\neg p) \vee q$	dupla negação, 1
4. $\neg p \rightarrow q$	implicação, 3
5. q	*modus ponens*, 4, 2

Note que na etapa 3 dessa prova, usamos uma das regras de equivalência (dupla negação) para fazer a substituição na fórmula. Isso é permitido: uma vez que $\neg(\neg p)$ é logicamente equivalente a p, ele pode tomar o lugar de p em qualquer fórmula.

1.5.2 Exercícios propostos

1 Complete a coluna das razões na sequência de prova seguinte. Certifique-se de indicar a que etapa(s) cada regra de dedução se refere.

Sentença	Razões
1. $p \wedge (q \vee r)$	dada
2. $\neg(p \wedge q)$	dada
3. $\neg p \vee \neg q$	
4. $\neg q \vee \neg p$	
5. $q \rightarrow \neg p$	
6. p	
7. $\neg(\neg p)$	
8. $\neg q$	
9. $q \vee r$	
10. $r \vee q$	
11. $\neg(\neg r) \vee q$	
12. $\neg r \rightarrow q$	
13. $\neg(\neg r)$	
14. r	
15. $p \wedge r$	

2 Justifique cada conclusão com uma regra de dedução.

a) Quem é artístico deve ser também criativo. Joselino não é criativo. Portanto, Joselino não é artístico.

b) Leonídio é atlético e também inteligente. Portanto, Leonídio é atlético.

c) Quem tem 18 anos de idade pode votar. Mafalda tem 18 anos de idade. Portanto, Mafalda pode votar.

3 Escreva uma sequência de prova para a seguinte asserção $\left.\begin{array}{l} p \\ p \rightarrow r \\ q \rightarrow \neg r \end{array}\right\} \Rightarrow \neg q$. Justifique cada etapa.

4 Escreva uma sequência de prova para a seguinte asserção $\left.\begin{array}{l} p \rightarrow q \\ p \wedge r \end{array}\right\} \Rightarrow q \wedge r$. Justifique cada etapa.

1.6 Lógica de Predicados

Quando definimos o que é uma sentença, dissemos que a frase da forma

$$x \text{ é par}$$

não é uma sentença, porque seus valores V /F dependem de x. A escrita matemática, no entanto, quase sempre lida com frases desse tipo; frequentemente expressamos ideias matemáticas em termos de uma variável desconhecida. Esta seção explica como estender nosso sistema formal de lógica para lidar com essas situações.

Definição 1.3: Predicado

Um *predicado* é uma frase declarativa cujos valores V/F dependem de uma ou mais variáveis.

Em outras palavras, um predicado é uma frase declarativa com variáveis que, após terem recebido valores específicos, transformam a frase em uma sentença.

Usamos a notação de função para denotar predicados. Por exemplo,

$$P(x) = \text{“}x \text{ é par”} \qquad \text{e}$$

$$Q(x, y) = \text{“}x \text{ é mais pesado que } y\text{”}$$

são predicados. A sentença $P(8)$ é verdadeira, enquanto a sentença $Q(pena, tijolo)$ é falsa.

Implícito em um predicado está o *domínio* (ou universo) de valores que a variável ou as variáveis podem assumir. Para $P(x)$, o domínio poderia ser de números inteiros; para $Q(x, y)$, o domínio poderia ser alguma coleção de objetos físicos. Em geral, vamos declarar o domínio juntamente com o predicado, a menos que já esteja claro pelo contexto.

Equações são predicados. Por exemplo, se $E(x)$ representa a equação

$$x^2 - x - 6 = 0,$$

então $E(3)$ é verdadeira e $E(4)$ é falsa. Consideramos equações como frases declarativas, em que o sinal = faz o papel de um verbo.

Sozinhos, os predicados não são sentenças porque contêm variáveis livres. Podemos transformá-los em sentenças subtituindo as variáveis por valores específicos do domínio; porém muitas vezes queremos obter uma sentença sem usar valores específicos.

Definição 1.4: Quantificador

Um *quantificador* modifica um predicado ao descrever se alguns ou todos os elementos do domínio satisfazem o predicado.

Vamos precisar somente de dois quantificadores: universal e existencial. O *quantificador universal* “para todo” é denotado por $\forall$. Então a sentença

$$(\forall x)P(x)$$

diz que $P(x)$ é verdadeira para todo x que está no domínio.

O *quantificador existencial* “existe” é denotado por $\exists$. A sentença

$$(\exists x)P(x)$$

diz que existe um elemento x do domínio tal que $P(x)$ é verdadeira; em outras palavras, $P(x)$ é verdadeira para *pelo menos um valor de x no domínio.* Por exemplo, se $E(x)$ é a equação $x^2 - x - 6 = 0$ no domínio dos números reais, então a expressão

$$(\exists x)E(x)$$

diz: "Existe algum número real x tal que $x^2 - x - 6 = 0$", ou de maneira mais simples, "A equação $x^2 - x - 6 = 0$ tem uma solução". A variável x não é mais uma variável livre, uma vez que o quantificador $\exists$ muda o papel que ele desempenha na sentença.

Se $Z(x)$ representa a equação $x \cdot 0 = 0$ nos números reais, a expressão

$$(\forall x)Z(x)$$

significa "Para todos os números reais x, vale que $x \cdot 0 = 0$". Novamente, essa é uma sentença sem variáveis livres, uma vez que o alcance de valores possíveis de x é claramente especificado.

Quando colocamos um quantificador em frente a um predicado, formamos uma *sentença quantificada.* Uma vez que o quantificador restringe o alcance de valores para as variáveis no predicado, a sentença quantificada é ou verdadeira ou falsa (mas não ambas). Nos exemplos anteriores, $(\exists x)E(x)$ e $(\forall x)Z(x)$ são ambas verdadeiras, enquanto a sentença

$$(\forall x)E(x)$$

é falsa, uma vez que alguns números reais não satisfazem a equação $x^2 - x - 6 = 0$.

O real poder da lógica de predicados vem de combinar quantificadores, predicados e os símbolos de lógica proposicional. Por exemplo, se quiséssemos afirmar que existe um número negativo que satisfaz a equação $x^2 - x - 6 = 0$, poderíamos definir um novo predicado

$$N(x) = \text{"}x \text{ é negativo"}.$$

Então a sentença

$$(\exists x)(N(x) \wedge E(x))$$

é traduzida para "Existe algum número real x tal que x é negativo e $x^2 - x - 6 = 0$."

Existem muitas formas diferentes de escrever sentenças quantificadas em português. Traduzir sentenças de uma forma para a outra, entre o português e a lógica de predicados, é uma habilidade que requer prática.

Exemplo 1.8

No domínio de todos os carros, sejam os predicados:

$$P(x) = \text{"}x \text{ consome pouco combustível."}$$
$$Q(x) = x \text{ é grande.}$$

então a sentença $(\forall x)(Q(x) \rightarrow \neg P(x))$ poderia ser literalmente traduzida para
"Para todos os carros x, se x é grande, então x não consome pouco combustível."

No entanto, uma tradução mais natural da mesma sentença é
"Todo carro grande consome muito combustível." ou
"Não existem carros grandes que consumam pouco combustível."
Se quiséssemos dizer o oposto, ou seja, que existem alguns carros grandes que consomem pouco combustível, poderíamos escrever a seguinte sentença:

$$(\exists x)(P(x) \wedge Q(x))$$

Daremos uma demonstração formal de que essa negação é correta no Exemplo 1.11.

O exemplo seguinte mostra como uma sentença matemática aparentemente simples rende uma fórmula um tanto complicada em lógica de predicados. O uso cuidadoso de predicados ajuda a revelar a estrutura lógica de uma afirmação matemática.

Exemplo 1.9

No domínio de todos os números inteiros, seja $P(x)$ = "x é par". A seguir, veja como podemos expressar o fato de que a soma de um número par com um número ímpar é ímpar:

$$(\forall x)(\forall(y)[(P(x) \wedge \neg P(y)) \rightarrow (\neg P(x+y))]$$

É claro que a tradução literal dessa sentença quantificada é
"Para todo número inteiro x e para todo número inteiro y, se x é par e y não é par, então $x+y$ não é par", mas normalmente dizemos algo informal como
"Um número par mais um número ímpar é ímpar".

Esse último exemplo usou dois quantificadores universais para expressar um fato sobre um par arbitrário x, y de números inteiros. O próximo exemplo mostra o que pode acontecer quando combinamos quantificadores universal e existencial na mesma sentença.

Exemplo 1.10

No domínio de todos os números reais, seja $G(x, y)$ o predicado "$x > y$". A sentença

$$(\forall y)(\exists x)G(x, y)$$

diz literalmente que "Para todos os números y, existe algum número x tal que $x > y$", ou, mais simples ainda, "Dado qualquer número y, existe algum número que é maior que y".
Essa sentença é claramente verdadeira: o número $y + 1$ é sempre maior que y, por exemplo.
No entanto, a sentença

$$(\exists x)(\forall y)G(x, y)$$

é traduzida literalmente como "Existe um número x tal que para todos os números y temos $x > y$". Em linguagem mais simples, essa sentença diz: "Existe algum número que é maior do que qualquer outro número." Essa sentença é claramente falsa, porque não existe um número maior que todos os outros.

A ordem dos quantificadores importa. Em ambas as sentenças, é feita uma afirmação de que x é maior que y. Na primeira sentença, primeiramente lhe é dado um número arbitrário y, e então a afirmação é que é possível achar algum x que seja maior que ele. No entanto, a segunda sentença afirma existir algum número x tal que, dado qualquer outro y, o maior dos dois será x. Na segunda sentença, você deve decidir o que é x antes de escolher y. Na primeira sentença, você escolhe y antes para em seguida escolher x.

A coisa mais importante que você precisa ser capaz de fazer com lógica de predicados é escrever a negação de uma sentença quantificada. Como em lógica proposicional, existem algumas equivalências formais que descrevem como a negação funciona.
A Tabela abaixo lista duas regras importantes para formar o oposto de uma sentença quantificada.

Equivalência	Nome
$\neg[(\forall x)P(x)] \iff (\exists x)(\neg P(x))$	negação universal
$\neg[(\exists x)P(x)] \iff (\forall x)(\neg P(x))$	negação existencial

É fácil ver o padrão formal dessas duas regras: para negar uma sentença quantificada, traga a negação para dentro do quantificador e troque o quantificador.

Vamos interpretar as regras de negação no contexto de um exemplo. No domínio de todas as pessoas, seja $L(x)$ o predicado "x é um mentiroso". A regra universal de negação diz que a negação para "Todas as pessoas são mentirosas" é "Existe uma pessoa que não é mentirosa". Em símbolos,

$$\neg[(\forall x)L(x)] \leftrightarrow (\exists x)(\neg L(x)).$$

Similarmente, a regra de negação existencial diz que a negação de "Existe um mentiroso" é "Não existem mentirosos".

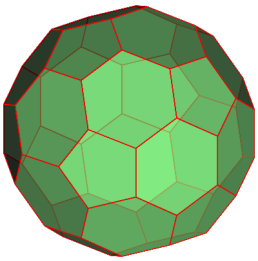
Exemplo 1.11

No Exemplo 1.8, discutimos o que a negação da sentença
"Todo carro grande consome muito combustível." deveria ser.
Podemos responder a essa questão negando a sentença formal $(\forall x)(Q(x) \rightarrow \neg P(x))$ usando uma sequência de equivalências de prova.
Vamos supor como dada a negação da sentença formal e deduzir uma sentença equivalente.

$$\begin{aligned}
& \neg[(\forall x)(Q(x) \rightarrow \neg P(x))] && \text{(sequência dada)} \\
\iff\ & (\exists x)\neg(Q(x) \rightarrow \neg P(x)) && \text{(negação universal)} \\
\iff\ & (\exists x)\neg(\neg Q(x) \vee \neg P(x)) && \text{(implicação)} \\
\iff\ & (\exists x)(\neg(\neg Q(x)) \wedge \neg(\neg P(x))) && \text{(lei de De Morgan)} \\
\iff\ & (\exists x)(Q(x) \wedge P(x)) && \text{(dupla negação)} \\
\iff\ & (\exists x)(P(x) \wedge Q(x)) && \text{(comutatividade)}
\end{aligned}$$

Note que o resultado concorda com a negação intuitiva que fizemos no Exemplo 1.8:
existe algum carro que é grande e que consome pouco combustível.

Considere o domínio constituído pelas faces do icosaedro truncado (também conhecido como bola de futebol):



Agora considere os predicados seguintes:

$$\begin{aligned}
P(x) &= \text{"}x \text{ é um pentágono"}. \\
H(x) &= \text{"}x \text{ é um hexágono"}. \\
B(x,y) &= \text{"}x \text{ faz fronteira com } y\text{"}.
\end{aligned}$$

Aqui dizemos que dois polígonos fazem fronteira um com o outro se eles compartilham uma aresta.

Questão 1: resolvida

Confirme que as observações seguintes são verdadeiras para qualquer icosaedro truncado:

a) Dois pentágonos nunca fazem fronteira um com o outro.

b) Todo pentágono faz fronteira com algum hexágono.

c) Todo hexágono faz fronteira com um outro hexágono.

Escreva essas sentenças em lógica de predicados e também as suas negações. Simplifique as sentenças negadas de modo que nenhum predicado permaneça dentro do escopo de uma negação. Traduza a sua sentença negada de volta para o português.

Solução: A formalização dessas sentenças se dá da seguinte forma.

① $(\forall x)(\forall y)((P(x) \wedge P(y)) \rightarrow \neg B(x, y))$

② $(\forall x)(P(x) \rightarrow (\exists y)(H(y) \wedge B(x, y)))$

③ $(\forall x)(H(x) \rightarrow (\exists y)(H(y) \wedge B(x, y)))$

Vamos negar ② e deixar as outras como exercício. Veja se você consegue descobrir as razões para cada equivalência.

$$
\begin{aligned}
& \neg[(\forall x)(P(x) \rightarrow (\exists y)(H(y) \wedge B(x, y)))] && \text{(sequência dada)} \\
\iff \quad & (\exists x)[\neg(P(x) \rightarrow (\exists y)(H(y) \wedge B(x, y)))] \\
\iff \quad & (\exists x)[\neg(\neg P(x) \vee (\exists y)(H(y) \wedge B(x, y)))] \\
\iff \quad & (\exists x)[\neg\neg(P(x) \wedge \neg(\exists y)(H(y) \wedge B(x, y)))] \\
\iff \quad & (\exists x)[\neg\neg(P(x) \wedge (\forall y)\neg(H(y) \wedge B(x, y)))] \\
\iff \quad & (\exists x)[(P(x) \wedge (\forall y)\neg(H(y) \wedge B(x, y)))] \\
\iff \quad & (\exists x)[(P(x) \wedge (\forall y)(\neg H(y) \vee \neg B(x, y)))] \\
\iff \quad & (\exists x)[(P(x) \wedge (\forall y)(H(y) \rightarrow \neg B(x, y)))]
\end{aligned}
$$

Essa última sentença diz que existe um x tal que x é um pentágono e, para qualquer y, se y é um hexágono, então x não faz fronteira com y. Em outras palavras, existe algum pentágono que não faz fronteira com hexágonos. Se você descobriu um sólido com essa propriedade, ele não poderia ser um icosaedro truncado.

Existem duas expressões que aparecem com frequência, e conhecer a lógica de predicados para essas expressões torna a tradução muito mais fácil. A primeira sentença é

Todo ⟨alguma coisa⟩ é ⟨alguma outra coisa⟩.

Por exemplo, "Todos os jogadores de beisebol são ricos", ou "Todas as ovelhas são brancas". Em geral, se $P(x)$ e $Q(x)$ são os predicados "x é ⟨alguma coisa⟩" e "x é ⟨alguma outra coisa⟩" respectivamente, então a expressão de lógica de predicados

$$(\forall x)(P(x) \rightarrow Q(x))$$

é traduzida como "Para todo x, se x é ⟨alguma coisa⟩, então x é ⟨alguma outra coisa⟩". Dito de forma mais simples, "Todos os x's com a propriedade ⟨alguma coisa⟩ devem ter a propriedade ⟨alguma outra coisa⟩," ou mais simples ainda, "Todo ⟨alguma coisa⟩ é ⟨alguma outra coisa⟩".

No domínio de todas as pessoas, se $R(x)$ representa "x é rico" e $B(x)$ representa "x é jogador de beisebol", então

$$(\forall x)(B(x) \rightarrow R(x))$$

é a sentença "Todos os jogadores de beisebol são ricos".

A segunda construção é da forma

Existe um ⟨alguma coisa⟩ que é ⟨alguma outra coisa⟩.

Por exemplo, "Existe um jogador de beisebol rico", ou "Existe uma ovelha branca". Essa expressão tem a seguinte forma em lógica de predicados:

$$(\exists x)(P(x) \wedge Q(x))$$

Note que isso se traduz literalmente como "Existe algum x tal que x é ⟨alguma coisa⟩ e x é ⟨alguma outra coisa⟩", que é o que queremos.

No domínio dos mamíferos, se $O(x)$ é o predicado "x é uma ovelha" e $F(x)$ é o predicado "x é branca", então

$$(\exists x)(F(x) \wedge O(x))$$

seria traduzida para "Existe uma ovelha branca".

Nota:

Você também poderia dizer "Existe uma ovelha de cor branca", "Algumas ovelhas são brancas", ou, até de uma forma mais esquisita, "Existe um mamífero branco que é uma ovelha". Todas essas sentenças significam a mesma coisa.

1.6.1 Exercícios propostos

1 No domínio dos números inteiros, seja $P(x, y)$ o predicado "$x \cdot y = 12$". Diga se cada uma das sentenças a seguir é verdadeira ou falsa:

a) $P(3, 4)$

b) $P(3, 5)$

c) $P(2, 6) \vee P(3, 7)$

d) $(\forall x)(\forall y)(P(x, y) \rightarrow P(y, x))$

e) $(\forall x)(\exists y)P(x, y)$

2 No domínio de todos os livros, considere os predicados seguintes:

$$H(x) = \text{"}x \text{ é pesado"}.$$
$$C(x) = \text{"}x \text{ é confuso"}.$$

Traduza as sentenças seguintes de lógica de predicados para o português cotidiano:

a) $(\forall x)(H(x) \rightarrow C(x))$

b) $(\exists x)(C(x) \wedge H(x))$

c) $(\forall x)(C(x) \vee H(x))$

d) $(\exists x)(H(x) \wedge \neg C(x))$

3 Considere os seguintes predicados no domínio de todas as plantas:

$$P(x) = \text{"}x \text{ é venenosa"}.$$
$$Q(x) = \text{"Jacinto comeu } x\text{"}.$$

Traduza as sentenças seguintes para a lógica de predicados.

a) Algumas plantas são venenosas.

b) Jacinto nunca comeu uma planta venenosa.

c) Existem algumas plantas não venenosas que Jacinto nunca comeu.

4 O domínio para este problema é uma coleção de números não especificados. Considere o predicado

$$P(x, y) = \text{"}x \text{ é maior que } y\text{"}.$$

a) "Traduza a sentença a seguir usando a lógica de predicados.

Para todo número existe um outro maior que ele.

b) Negue a sua expressão da parte a) e simplifique-a de forma que não restem quantificadores dentro do escopo de uma negação.

c) Traduza a sua expressão da parte b) para um português compreensível. Não use variáveis na sua tradução para o português.

5 Qualquer equação ou desigualdade com variáveis é um predicado no domínio dos números reais. Diga se cada uma das sentenças seguintes é verdadeira ou falsa:

a) $(\forall x)(x^2 > x)$

b) $(\exists x)(x^2 - 2 = 1)$

c) $(\exists x)(x^2 + 2 = 1)$

d) $(\forall x)(\exists y)(x^2 + y = 4)$

e) $(\exists y)(\forall x)(x^2 + y = 4)$

1.7 Conectivos Lógicos no Mundo Real

Os programas de busca na Internet permitem a exploração de recursos imensos disponíveis, mas um pouco de cuidado na sua pesquisa pode ajudar a chegar ao resultado desejado mais rapidamente. Por exemplo, se pesquisar

carros usados

em um programa de busca, você pode obter de volta referências na rede de qualquer página contendo a palavra carros ou a palavra usados; isso poderia incluir antiquários e páginas contendo os últimos resultados das corridas. Se você escrever

"carros usados"

entre aspas, na maior parte dos programas de busca, isso restringiria a busca às páginas contendo exatamente essa frase. A maior parte dos programas de busca permite que você coloque uma expressão usando conectivos lógicos em sua pesquisa, o que ajuda a tornar a pesquisa ainda mais específica. Para diminuir ainda mais sua pesquisa sobre carros usados, você poderia colocar, por exemplo,

"carros usados" E (Ford OU Gurgel)

Isso tenderia a limitar sua pesquisa a lugares que mencionam marcas particulares de carros usados, embora, ainda assim, você possa terminar com um link para a Agência Pirata de Empréstimos Jaime Gurgel, que empresta dinheiro para comprar carros usados. A pesquisa

"carros usados" E (Ford OU Gurgel) E NÃO caminhões

eliminaria os lugares que mencionam caminhões. Muitos programas de busca usam + (um sinal de mais) no lugar de E (ou AND) e − (um sinal de menos) no lugar de E NÃO (ou AND NOT).

Os conectivos lógicos E (AND), OU (OR) e NÃO (NOT) também estão disponíveis em muitas linguagens de programação, assim como em calculadoras gráficas programáveis. Esses conectivos, de acordo com as tabelas verdade que definimos, agem em combinações de expressões verdadeiras ou falsas para produzir um valor lógico final. Tais valores lógicos fornecem a capacidade de decisão fundamental ao fluxo de controle em programas de computadores. Assim, em uma ramificação condicional de um programa, se o valor lógico da expressão condicional for verdadeiro, o programa executará a seguir um trecho de seu código; se o valor for falso, ele executará um trecho diferente de seu código. Se a expressão condicional for substituída por outra expressão equivalente mais simples, o valor lógico da expressão, e, portanto, o fluxo de controle do programa, não será afetado, mas o novo código será mais fácil de ser entendido e poderá ser executado mais rapidamente.

Exemplo 1.12

Considere uma proposição em um programa de computador da forma

```
1 se ((FluxoSaída > FluxoEntrada) e não ((FluxoSaída > FluxoEntrada) e (Pressão < 1000)))
    então
2   faça AlgumaCoisa;
3 senão
4   faça OutraCoisa
```

É fácil ver que esse proposição condicional tem a forma $A \wedge \neg(A \wedge B)$ em que A = "FluxoDeSaída > FluxoDeEntrada" e B = "Pressão < 1000". Essa expressão pode ser simplificada substituindo-a por outra equivalente.

$$
\begin{aligned}
A \wedge \neg(A \wedge B) &\iff A \wedge (\neg A \vee \neg B) && \text{(lei de De Morgan)} \\
&\iff (A \wedge \neg A) \vee (A \wedge \neg B) && \text{(distributividade)} \\
&\iff F \vee (A \wedge \neg B) && \text{(contradição)} \\
&\iff (A \wedge \neg B) && \text{(propriedade da identidade)}
\end{aligned}
$$

A proposição pode, então, ser escrita na forma

```
1 se ((FluxoSaída > FluxoEntrada) e não (Pressão < 1000)) então
2 |  faça AlgumaCoisa;
3 senão
4 |  faça OutraCoisa
```

Finalmente, as tabelas verdade para a conjunção, a disjunção e a negação são implementadas por dispositivos eletrônicos chamados "portas lógicas" (porta lógica E ou AND, porta lógica OU ou OR, inversor, respectivamente) como mostrado no Exercício 8 da Seção 1.3.1, que são os módulos fundamentais na construção dos circuitos nos computadores.

1.7.1 Algoritmos e Lógica de Programação

Para testar se uma proposição é uma tautologia, sempre podemos construir sua tabela verdade. Para n argumentos, precisaremos de 2^n linhas para a tabela verdade. Suponha, entretanto, que o conectivo principal é uma implicação, ou seja, que a proposição tem a forma $P \rightarrow Q$. Então, podemos usar um procedimento mais rápido do que construir uma tabela verdade para decidir se $P \rightarrow Q$ é, ou não, uma tautologia. Vamos supor que $P \rightarrow Q$ não é uma tautologia e ver se isso nos leva a uma situação impossível. Se esse for o caso, então a hipótese de que $P \rightarrow Q$ não é uma tautologia também é impossível, e $P \rightarrow Q$ vai ter que ser mesmo uma tautologia. Supor que $P \rightarrow Q$ não é uma tautologia é o mesmo que dizer que ela pode assumir o valor falso, e, pela tabela verdade para implicação, $P \rightarrow Q$ só é falsa quando P é verdadeira e Q é falsa. Designando os valores lógicos verdadeiro para P e falso para Q, determinamos os possíveis valores lógicos para as sentenças que constituem P e Q. Continuamos a atribuição de valores lógicos dessa forma até que todas as sentenças tenham um valor lógico. Se, por esse processo, forem atribuídos a alguma sentença os valores lógicos verdadeiro e falso ao mesmo tempo, teremos uma situação impossível, logo $P \rightarrow Q$ terá que ser uma tautologia. Caso contrário, encontraremos uma maneira de tornar a proposição atribuída falsa, e, portanto, ela não é uma tautologia.

O que descrevemos é um conjunto de instruções – um procedimento – para executar a tarefa de determinar se $P \rightarrow Q$ é, ou não, uma tautologia. Esse procedimento pode ser executado mecanicamente seguindo-se as instruções; em um tempo finito, teremos uma resposta. Em ciência da computação, tal procedimento é chamado um *algoritmo*.

Definição 1.5: algoritmo

Um *algoritmo* é um conjunto de instruções que podem ser executadas mecanicamente em um tempo finito de modo a resolver algum problema.

Algoritmos constituem o cerne da ciência da computação, e teremos mais a dizer sobre eles ao longo deste livro. Você, provavelmente, já sabe que a tarefa principal ao escrever um programa de computador para resolver um problema consiste em desenvolver um algoritmo para produzir a solução do problema.

Os algoritmos são, muitas vezes, descritos de maneira intermediária entre uma descrição puramente verbal em um parágrafo (como fizemos anteriormente para decidir se $P \to Q$ era uma tautologia) e um programa de computador (que, se fosse executado, realizaria, de fato, os passos do algoritmo) escrito em uma linguagem de programação.

Essa forma intermediária de escrever algoritmos é conhecida como **pseudocódigo**. Um algoritmo escrito em pseudocódigo não deve ser difícil de entender, mesmo se você não souber nada sobre programação de computadores. A única coisa a observar sobre o pseudocódigo utilizado neste livro é que linhas que começam com duas barras inclinadas (//) são comentários, não fazendo parte do algoritmo propriamente dito.

Abaixo vemos em pseudocódigo o algoritmo para determinar se $P \to Q$ é, ou não, uma tautologia.

Algoritmo TESTATAUTOLOGIA

```
   // Dadas P e Q, decide se P → Q é, ou nao, uma tautologia.
   // Suponha que P → Q nao é uma tautologia
1  P = verdadeira // atribui V a P
2  Q = falsa // atribui F a Q
3  repita
4  |   para cada sentença composta à qual já tenha sido atribuída um valor lógico, atribua os
   |     valores lógicos determinados para cada uma de suas componentes
5  até todas as componentes da proposição terem valores lógicos
6  se alguma componente da proposição tiver dois valores lógicos diferentes
       // contradiçao, a hipótese é falsa
7  então
8  |   escreva ("P → Q é uma tautologia.")
9  senão
10 |   escreva ("P → Q não é uma tautologia.") // existe uma maneira de tornar
   |      P → Q falsa
```

O algoritmo atribui primeiro os valores lógicos "verdadeiro" a P e "falso" a Q, de acordo com a hipótese de que $P \to Q$ não é uma tautologia. O algoritmo, então, entra em um laço (ou ciclo), em que uma sequência de passos é repetida até que determinada condição seja satisfeita. Dentro do laço, atribuem-se valores lógicos às componentes cada vez menores das proposições originais P e Q até que todas as letras de proposição tenham valores lógicos associados. Então o algoritmo testa se ocorreu alguma contradição e escreve a informação sobre se $P \to Q$ é, ou não, uma tautologia.

Exemplo 1.13

Considere a proposição $(A \to B) \to (\neg B \to \neg A)$. Essa proposição tem a forma necessária para se usar o algoritmo TestaTautologia, ou seja, é da forma $P \to Q$, em que P é $(A \to B)$ e Q é $(\neg B \to \neg A)$. Seguindo o algoritmo, atribuímos primeiro os valores lógicos $(A \to B)$ verdadeira e $(\neg B \to \neg A)$ falsa Entrando no laço, a atribuição de falsa à proposição composta $(\neg B \to \neg A)$ determina outras atribuições, a saber,

$$\neg B \text{ verdadeira e } \neg A \text{ falsa}$$

ou

$$B \text{ falsa e } A \text{ verdadeira.}$$

Trabalhando agora com P, A verdadeira e $A \to B$ verdadeira determina a atribuição B verdadeira

(por *modus ponens*). Agora todas as letras de proposição têm valor lógico, ou seja:

$$\overbrace{(\underbrace{A}_{\text{A=V}} \rightarrow \underbrace{B}_{\text{B=V}})}^{\text{V}} \rightarrow \overbrace{(\underbrace{\neg B}_{\text{B=F}} \rightarrow \underbrace{\neg A}_{\text{A=V}})}^{\text{F}}$$

Isso encerra o laço. No último passo do algoritmo, B tem, ao mesmo tempo, os valores lógicos V e F, logo o algoritmo decide que $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$ é uma tautologia. De fato, você pode construir uma tabela verdade e verificar isso (se tiver paciência).

O algoritmo TestaTautologia decide se uma proposição de determinada forma, ou seja, cujo conectivo principal é $\rightarrow$, é uma tautologia. No entanto, o processo de construção de uma tabela verdade, seguido do exame dos valores lógicos na última coluna, também constitui um algoritmo para decidir se uma proposição arbitrária é uma tautologia. Esse segundo algoritmo é mais poderoso, já que resolve um problema mais geral, mas o algoritmo TestaTautologia é, geralmente, mais rápido para aquelas proposições a que se aplica.

1.7.2 Exercícios propostos

1 Considere o seguinte pseudocódigo:

```
1 repita
2 |  i = 1 // primeiro valor x
3 |  leia o valor de x
4 |  se ((x < 5, 0) e (2x < 10, 7)) ou
  |   (√5x > 5, 1) então
5 |  |  escreva o valor de x
6 |  aumente i em 1 // próximo valor
7 até i > 5
```

Os valores de entrada são 1,0; 5,1; 2,4; 7,2 e 5,3. Quais são os valores de saída?

2 Reescreva o pseudocódigo a seguir com uma expressão condicional mais simples, em que a função *impar*(*n*) tem o valor lógico verdadeiro se *n* for ímpar.

```
1 se não ((Valor1 < Valor2) ou
  ímpar(Número))
2 ou (não(Valor1 < Valor2) e
  ímpar(Número)) então
3 |  proposição1
4 senão
5 |  proposição2
```

3 Você quer que seu programa execute a proposição1 quando A for falsa, B for falsa e C for verdadeira e que execute a proposição2 nos outros casos. Você escreveu:

```
1 se não (A e B) e C então
2 |  proposição1
3 senão
4 |  proposição2
```

Esse algoritmo faz o que você quer?

4 Use o algoritmo TestaTautologia para provar que as expressões a seguir são tautologias.

a) $[\neg B \wedge (A \rightarrow B)] \rightarrow \neg A$

b) $[(A \rightarrow B) \wedge A] \rightarrow B$

c) $(A \vee B) \wedge \neg A \rightarrow B$

5 Um anúncio de um restaurante em um clube exclusivo em Honolulu diz

"Apenas sócios e não sócios".

Dê duas interpretações possíveis para essa afirmação.

Capítulo 2

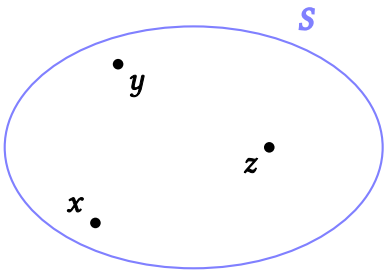
Pensamento Estrutural

Acreditamos que o leitor já teve contato com os conceitos básicos da teoria dos conjuntos, como elemento, união, intersecção, etc.. Nesta seção vamos revisar esses conceitos. Embora seja possível desenvolver a teoria de conjuntos de maneira axiomática, como foi feito por George Cantor (1845–1918) e Ernest Zermelo (1871–1953), a abordagem informal apresentada é suficiente para nossos propósitos.

2.1 Conjuntos

A forma mais simples de descrever uma coleção de objetos relacionados é como um conjunto. Pense no conjunto S como um recipiente em que um objeto x é alguma coisa que S contém.

Escrevemos $x \in S$ para denotar que x está contido em S. Também dizemos que "x é um membro de S, "x é um elemento de S", ou, mais simplesmente ainda, "x está em S". Se houver mais elementos, podemos escrever $x, y, z \in S$ para denotar que estes três elementos estão no mesmo conjunto como na figura ao lado.



Podemos descrever exemplos de conjuntos ao listar os elementos no conjunto ou descrever as propriedades que um elemento do conjunto possui. Para dizer que o conjunto S consiste nos elementos $x_1, x_2, \ldots, x_n$, escrevemos

$$S = \{x_1, x_2, \ldots, x_n\}.$$

Suponha que p seja uma propriedade que alguns dos elementos do conjunto S possuam. Podemos descrever o conjunto de todos os elementos de S que têm a propriedade p como

$$\{x \in S \mid x \text{ tem propriedade } p\}.$$

Algumas vezes essa notação é chamada de "construtor de conjuntos", porque explica como construir uma lista de todos os elementos de um conjunto.

Exemplo 2.1

Seja $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Então $2 \in A$ e $9 \notin A$. Se

$$B = \{x \in A \mid x \text{ é ímpar }\},$$

então os elementos de B são 1, 3, 5, 7.

Exemplo 2.2

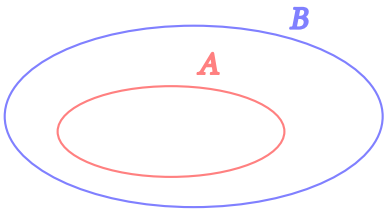
Existem alguns conjuntos mais comuns que recebem nomes específicos. O conjunto dos números inteiros é denotado por $\mathbb{Z}$, e o conjunto dos números inteiros positivos (ou números naturais) é escrito como $\mathbb{N}$. Note que $0 \in \mathbb{Z}$, mas $0 \notin \mathbb{N}$. Usamos $\mathbb{R}$ para o conjunto de números reais e $\mathbb{Q}$ para o conjunto de números racionais, ou seja, o conjunto de todas as frações $\frac{a}{b}$, em que a e b são números inteiros, e $b \neq 0$.

Exemplo 2.3

Seja P o conjunto de todos os polígonos. Então P contém todos os triângulos, quadrados, pentágonos, etc. Se c é um círculo, então $c \notin P$. Poderíamos descrever o conjunto H de todos os hexágonos em uma notação construtora de conjunto como

$$H = \{x \in P \mid x \text{ tem seis lados}\}.$$

A linguagem dos conjuntos é útil para descrever grupos de objetos que estão relacionados por alguma propriedade em comum. Muitas vezes uma propriedade implica outra; por exemplo, todos os números inteiros são números reais. Em termos de conjuntos, isso significa que o conjunto $\mathbb{Z}$ está contido no conjunto $\mathbb{R}$. Em geral, a afirmação do predicado lógico de que



$$(\forall x)(x \in A \rightarrow x \in B) \tag{2.1}$$

é escrita como $A \subseteq B$, e dizemos que "A está contido em B", ou "A é um subconjunto de B", ou "B contém A". Podemos expressar essa relação de forma pictórica usando o *diagrama de Venn* acima.

Exemplo 2.4

No Exemplo 2.1, $B \subseteq A$. Notamos que $\mathbb{Z} \subseteq \mathbb{R}$ no Exemplo 2.2, e também poderíamos dizer que $\mathbb{N} \subseteq \mathbb{Z}$, $\mathbb{Z} \subseteq \mathbb{Q}$, e $\mathbb{Q} \subseteq \mathbb{R}$. Essa cadeia de relações também poderia ser escrita como

$$\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}.$$

Exemplo 2.5

O conjunto vazio $\varnothing$ é o conjunto que não contém elementos. Portanto, o conjunto vazio é um subconjunto de qualquer conjunto, ou seja, $\varnothing \subseteq X$ para todo X. Isso acontece porque a sentença $x \in \varnothing$ é falsa para todo x, de modo que a implicação

$$(\forall x)(x \in \varnothing \rightarrow x \in X)$$

deve ser verdadeira. (Lembre-se da tabela verdade para o conectivo "$\rightarrow$" ao final da Seção 1.2.1.)

2.1.1 Novos Conjuntos a Partir de Antigos

Os conjuntos descrevem relacionamentos, mas a linguagem dos conjuntos também pode descrever a lógica de como as coisas estão relacionadas. Já vimos um exemplo desses antes: a sentença 2.1 mostra como interpretar o símbolo $\subseteq$ nos termos de uma sentença de predicado lógico contendo o conectivo $\rightarrow$; a relação de inclusão tem a lógica de uma implicação. Os conectivos $\vee$, $\wedge$, $\neg$ e $\leftrightarrow$ também têm suas representações na teoria dos conjuntos.

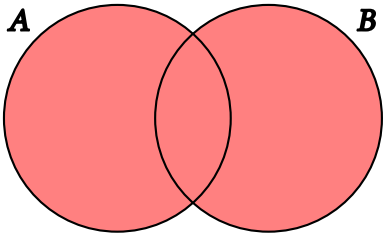
A *união* $A \cup B$ de dois conjuntos A e B é o conjunto que contém todos os elementos de A e B juntos. Um elemento pertence à união de dois conjuntos A e B se o elemento está em A, ou em B, ou em ambos. Na notação de construção dos conjuntos,

$$A \cup B = \{x \mid (x \in A) \vee (x \in B)\}.$$

Isso se traduz para a seguinte regra de equivalência em lógica de predicados.

$$(\forall x)[x \in A \cup B \Leftrightarrow (x \in A) \vee (x \in B)].$$

A sentença $x \in A \cup B$ é logicamente equivalente à sentença $(x \in A) \vee (x \in B)$. Esse fato é importante quando escrevemos demonstrações; uma dessas sentenças pode sempre ser substituída por outra.

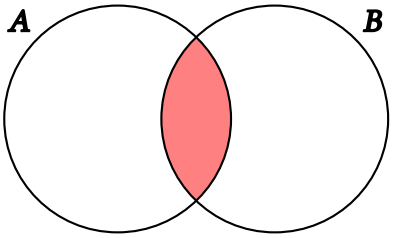


Pense em "união" como o homólogo em teoria dos conjuntos do conectivo lógico "ou". No diagrama de Venn ao lado, $A \cup B$ é a área sombreada.

A *interseção* $A \cap B$ é o conjunto que contém todos os elementos que A e B têm em comum. Para que um elemento esteja na interseção de A e B, o elemento deve estar em ambos os conjuntos. Portanto, escrevemos a interseção como

$$A \cap B = \{x \mid (x \in A) \wedge (x \in B)\}$$

na notação de construção de conjuntos. Isso implica a seguinte equivalência lógica para todo x.

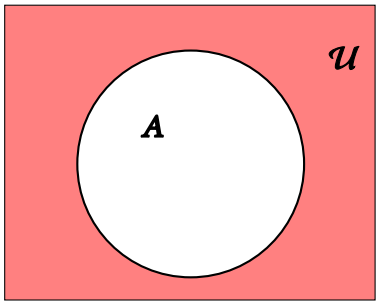


$$x \in A \cap B \Leftrightarrow (x \in A) \wedge (x \in B).$$

No diagrama de Venn ao lado, a área sombreada representa $A \cap B$.

Nas sentenças de lógica de predicados anteriores está implícito um domínio, ou *conjunto universo* $\mathcal{U}$, do qual todo conjunto é um subconjunto. Se sabemos o que é o domínio $\mathcal{U}$, podemos falar do *complemento* A^c de A, que é o conjunto

$$A^c = \{x \in \mathcal{U} \mid x \notin A\}.$$



Geralmente desenhamos $\mathcal{U}$ como um grande retângulo, de maneira que a área sombreada a seguir representa A^c.

Note que também poderíamos escrever $A^c = \{x \in \mathcal{U} \mid \neg(x \in A)\}$ para tornar explícito o uso do conectivo $\neg$.

Dizemos que dois conjuntos são idênticos se eles têm os mesmos elementos. Portanto, a sentença "$A = B$" é traduzida para a seguinte sentença em lógica de predicados:

$$(\forall x)(x \in A \leftrightarrow x \in B).$$

Essa tradução é importante quando se trata de demonstrar que dois conjuntos são idênticos. Uma prova de que $A = B$ geralmente consiste em duas provas diretas: dado $x \in A$, prove que $x \in B$, e ao contrário, dado $x \in B$, prove que $x \in A$. Em outras palavras, mostrar que $A = B$ corresponde a mostrar que $A \subseteq B$ e $B \subseteq A$.

Exemplo 2.6

Sejam dados os conjuntos a seguir:

$F = \{n \in \mathbb{Z} \mid n = 4k \text{ para algum inteiro } k\}$

$E = \{n \in \mathbb{Z} \mid n \text{ é par }\}$

Prove que $F \subseteq E$.

Solução: Seja $x \in F$ um elemento qualquer. Pela definição de F, temos $x = 4k$ para algum número inteiro k. Podemos escrever esta equação como $x = 2(2k)$, desse modo x é par. Assim, $x \in E$. Como tomamos um elemento qualquer, isso vale para todo o conjunto F, portanto $F \subseteq E$.

Em geral, qualquer sentença envolvendo conjuntos e os símbolos $\cap$, $\cup$, $\subseteq$, $=$ e ^{c} é traduzida para uma sentença lógica usando os conectivos $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ e $\neg$, respectivamente. Dessa forma todo o trabalho que fizemos em lógica proposicional e de predicados no capítulo anterior será compensado quando lidarmos com conjuntos.

Pense nos símbolos $\cap, \cup, \subseteq, =$ e ^{c} como ferramentas para se fazerem novos conjuntos a partir de outros mais antigos. Por exemplo, dados dois conjuntos, A e B, podemos construir um novo conjunto $A \cap B$, consistindo em todos os elementos que os dois conjuntos têm em comum. Também podemos tomar dois conjuntos A e B e formar o *produto cartesiano* de $A \times B$. O produto cartesiano é apenas o conjunto de todos os pares ordenados em que o primeiro item pertence ao primeiro conjunto e o segundo item pertence ao segundo conjunto. Formalmente,

$$A \times B = \{(a, b) \mid a \in A \text{ e } b \in B\}.$$

Também podemos ter trios ordenados, quartetos e assim por diante. O conjunto

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) \mid a_i \in A_i\}.$$

é o conjunto de n-uplas em que o i-ésimo item vem do conjunto A_i.

Dois pares ordenados são iguais se, e somente se, suas partes correspondentes são iguais. Em outras palavras, $(a, b) = (c, d) \Leftrightarrow a = c$ e $b = d$.

Você já está familiarizado com o produto cartesiano $\mathbb{R} \times \mathbb{R}$, o conjunto de todos os pares ordenados dos números reais. Esse conjunto é o plano cartesiano usual da álgebra ensinada no colégio.

Uma última construção que algumas vezes pode ser útil é o conjunto das partes de um conjunto S, o qual é denotado por $\mathcal{P}(S)$. O conjunto das partes de S é o conjunto de todos os subconjuntos de S:

$$\mathcal{P}(S) = \{X \mid X \subseteq S\}.$$

Note que o conjunto vazio $\varnothing$ é um membro de $\mathcal{P}(S)$, não importa o que seja S, porque o conjunto vazio é um subconjunto de qualquer conjunto.

Exemplo 2.7

Suponha que você queira formar um grupo de estudos com alguns dos outros alunos da sua turma. Se S é o conjunto de todos os alunos na sua turma, então $\mathcal{P}(S)$ é o conjunto de todos os grupos de estudos possíveis que você poderia formar. (O conjunto vazio representaria a decisão de não formar grupo de estudo algum!)

Exemplo 2.8

Sejam $A = \{1, 2, 3, 4, 5\}, B = \{4, 5, 6, 7, 8\}$, e suponha que o conjunto universo é $\mathcal{U} = \{1, 2, \ldots, 10\}$. Então

$$\begin{aligned} A \cup B &= \{1, 2, \ldots, 8\} \\ A \cap B &= \{4, 5\} \\ B^c &= \{1, 2, 3, 9, 10\} \\ (A \cap B) \times A &= \{(4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)\} \\ \mathcal{P}(A \cap B) &= \{\varnothing, \{4\}, \{5\}, \{4, 5\}\} \end{aligned}$$

Note que, enquanto A e B são conjuntos de números, os conjuntos que você constrói a partir de A e B podem ter elementos de diferentes tipos. Por exemplo, $(A \cap B) \times A$ acima é um conjunto de pares ordenados, e $\mathcal{P}(A \cap B)$ é um conjunto composto por conjuntos.

Os conjuntos que contêm um número finito de elementos são chamados de *conjuntos finitos*. É conveniente denotar o número de elementos em um conjunto S finito por $|S|$. Por exemplo, se S é o conjunto que contém todos os membros do Senado brasileiro, então $|S| = 81$.

Uma regra prática para lidar com tamanhos de conjuntos é o *princípio da inclusão-exclusão.*

$$|A \cup B| = |A| + |B| - |A \cap B| \tag{2.2}$$

Se olharmos para os diagramas de Venn para $A \cup B$ e $A \cap B$, no começo dessa seção, fica fácil visualizarmos porque essa regra é válida. Ao contar os elementos de $A \cup B$ contando os elementos de A e B e somando os resultados, contaríamos duas vezes os elementos que estão em $A \cap B$. Por esse motivo, "$-|A \cap B|$" aparece ao lado direito da Equação (2.2). Essa equação pode ajudar a organizar problemas de contagem envolvendo conjuntos com elementos em comum.

Exemplo 2.9

Em alguma universidade de São Paulo, para se tornar membro da Sociedade dos Mestres do Universo, é obrigatório ter conceito geral A na universidade ou ou pelo menos 900 pontos no ENEM. Dos 11 membros da sociedade, 8 fizeram pelo menos 900 pontos no ENEM, e 5 tiveram conceito geral A na universidade. Quantos membros tiveram tanto o mínimo de 900 pontos no ENEM quanto conceito geral A na universidade?

Solução: Seja A o conjunto de membros com conceito geral A, e seja B o conjunto de membros com pelo menos 900 pontos no ENEM. Então $A \cap B$ é o conjunto de membros com ambos os requisitos. Pelo princípio da inclusão-exclusão (Equação (2.2)),

$$11 = 5 + 8 - |A \cap B|$$

logo existem dois membros com ambos os requisitos.

2.1.2 Exercícios Propostos

1 Desenhe diagramas de Venn para ilustrar as regras de De Morgan para conjuntos.

2 Desenhe um diagrama de Venn para mostrar a região $A \cap B^c$. Essa região também é denotada por $A \setminus B$, e é chamada de *conjunto diferença*, por motivos óbvios.

3 Sejam A $= \{2, 3, 4\}$, $B = \{3, 4, 5, 6\}$, e suponha que o conjunto universo seja $\mathcal{U} = \{1, 2, \ldots, 9\}$. Liste todos os elementos dos conjuntos a seguir.

a) $(A \cup B)^c$
b) $(A \cap B) \times A$
c) $\mathcal{P}(B \setminus A)$
d) $(A \setminus B) \cup (B \setminus A)$

4 Sejam dados os conjuntos a seguir:

G = o conjunto de todos os cidadãos bons.
C = o conjunto de todas as pessoas caridosas.
P = o conjunto de todas as pessoas gentis.

Escreva a sentença "Toda pessoa que é caridosa e gentil é um bom cidadão" na linguagem da teoria dos conjuntos.

5 Considere os conjuntos a seguir. O conjunto universo para este problema é $\mathbb{N}$.

A = O conjunto de todos os números pares.
B = O conjunto de todos os números primos.
C = O conjunto de todos os quadrados perfeitos.
D = O conjunto de todos os múltiplos de 10.

Usando apenas os símbolos 3, A , B , C, D, $\mathbb{N}$, $\in$, $\subseteq$, $=$, $\neq$, $\cap$, $\cup$, $\times$, ^{c}, $\varnothing$, escreva as seguintes sentenças em notação de conjuntos.

a) Nenhum dos quadrados perfeitos é número primo.

b) Todos os múltiplos de 10 são números pares.

c) O número 3 é um número primo que não é par.

d) Se você pegar todos os números primos, todos os números pares, todos os quadrados perfeitos e todos os múltiplos de 10, você ainda não terá todos os números naturais.

6 Considere os conjuntos a seguir. O conjunto universo $\mathcal{U}$ para este problema é o conjunto de todas as pessoas residentes na índia.

A = O conjunto de todas as pessoas que falam inglês.
B = O conjunto de todas as pessoas que falam hindi.
C = O conjunto de todas as pessoas que falam urdu.

Expresse os seguintes conjuntos usando símbolos da teoria de conjuntos.

a) Residentes na índia que falam inglês, hindi e urdu.

b) Residentes na índia que não falam inglês, hindi ou urdu.

c) Residentes na índia que falam inglês, mas não falam hindi ou urdu.

7 Se $|A \cup B| = 20$, $|A| = 10$ e $|B| = 15$, encontre $|A \cap B|$. Faça um diagrama.

8 Se $|A \cup B| = 10$, $|A| = 8$ e $|A \cap B| = 4$, quantos elementos tem o conjunto B?

9 Em uma classe de 40 alunos, todos tem ou um *piercing* no nariz ou um *piercing* na orelha. O professor pede para que todos os alunos com *piercing* no nariz levantem as mãos. Nove mãos se levantam. Em seguida o professor pede que todos com *piercing* na orelha façam o mesmo. Dessa vez, 34 mãos se levantaram. Quantos alunos têm *piercings* tanto na orelha quanto no nariz?

10 Seja $S = \{a, b, c\}$. Escreva todos os elementos dos conjuntos a seguir.

a) $S \times S$

b) $\mathcal{P}(S)$

11 Liste todos os elementos de $\mathcal{P}(\mathcal{P}(\{1\}))$.

2.2 Funções

Você provavelmente já viu funções antes. Por exemplo, em álgebra no colégio, você aprendeu a representar graficamente funções como

$$f(x) = x^2 - 3x + 2 \tag{2.3}$$

e a fazer vários cálculos. Mas esse é um tipo muito específico de função; nesta seção iremos olhar as funções de uma perspectiva mais geral, usando a linguagem dosconjuntos.

Assim como os conjuntos, as funções descrevem relações matemáticas. Nesse tipo de relação, o valor de um objeto é completamente determinado pelo valor de outro. Na Equação (2.3), o valor de $f(x)$ fica fixado uma vez que sabemos qual o valor de x.

2.2.1 Definição e Exemplos

Definição 2.1: função

Uma *função* que vai de um conjunto X para um conjunto Y é uma regra precisa que atribui um único elemento de Y a cada elemento de X. Se f é tal função, escrevemos

$$f : X \to Y$$

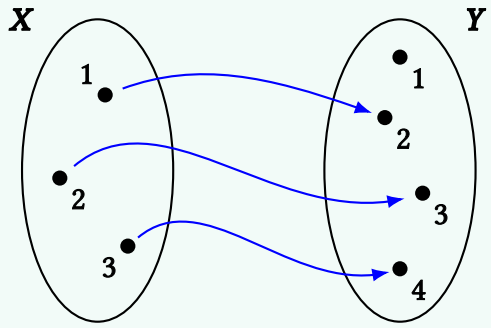
e denotamos o elemento de Y atribuído a $x \in X$ por $f(x)$. O conjunto X é chamado de *domínio* da função, e o conjunto Y é chamado de *contradomínio*.

Exemplo 2.10

Sejam $X = \{1, 2, 3\}$ e $Y = \{1, 2, 3, 4\}$. A fórmula $f(x) = x + 1$ define uma função $f : X \to Y$.

Para essa função, $f(1) = 2$, $f(2) = 3$ e $f(3) = 4$.

A figura ao lado mostra uma forma de representar graficamente essa associação entre os elementos.



Exemplo 2.11

A fórmula $f(x) = x^2 - 3x + 2$ define uma função $f : \mathbb{R} \to \mathbb{R}$.

Exemplo 2.12

Seja W o conjunto de todas as palavras deste livro, e seja L o conjunto de todas as letras no alfabeto. Defina a função $f : W \to L$ tomando $f(w)$ igual à primeira letra da palavra w. Note que a escolha da palavra w determina completamente $f(w)$, a primeira letra na palavra.

Exemplo 2.13

Uma sentença proposicional em duas variáveis define a função

$$z : \{V, F\} \times \{V, F\} \to \{V, F\}$$

onde $z(A, B)$ é o valor lógico da sentença quando as variáveis têm valores lógicos A e B. Por exemplo, a sentença $A \vee B$ define a função cujos valores são dados pela tabela ao lado.

A	B	$z(A,B)$
V	V	V
V	F	V
F	V	V
F	F	F

Exemplo 2.14

Seja F o conjunto de todos os conjuntos finitos não vazios de números inteiros, de modo que $F \subseteq \mathcal{P}(\mathbb{Z})$. Defina a função

$$s : F \to \mathbb{Z}$$

tomando $s(X)$ como a soma de todos os elementos de X. Por exemplo, $s(\{1, 2, 3\}) = 6$.

Uma outra palavra para função é *mapa* (alguns autores também usam o termo *aplicação*). Isso reflete o pensamento de que uma função é uma maneira de descrever uma rota de um conjunto a outro. No Exemplo 2.14, descrevemos uma forma de ir de um conjunto finito de inteiros para um número inteiro particular, ou seja, descrevemos um mapeamento a partir do conjunto de todos os conjuntos finitos de inteiros para o conjunto dos inteiros. Essa função associa ao conjunto $\{1, 2, 3\}$ o inteiro 6.

Isso ilustra um vasto leque de opções, já que podemos considerar conjuntos não apenas de números como também de listas de valores, de instruções, etc.

A notação $f(x)$ sugere o aspecto mais importante da definição da função: o valor da função é completamente determinado pelo objeto que você "insere" na função. Mais precisamente, a condição de que uma função deve ser *bem definida* significa que $f(x)$ tem um valor para cada x do domínio, e

$$a = b \Rightarrow f(a) = f(b)$$

para todo a e b no domínio.

Uma função $f : X \to Y$ pode falhar em ser bem definida de duas maneiras:

☞ algum x no domínio X pode falhar em ter um y no contradomínio ao qual se associa, ou

☞ algum x no domínio pode ser associado (ambiguamente) a dois y's diferentes no contradomínio.

Exemplo 2.15

Seja P o conjunto de todas as pessoas (mortas ou vivas). Seja

$$m : P \to P$$

tal que $m(x)$ é a mãe biológica de x. Temos que fazer suposições biológicas razoáveis para considerar esta uma função bem definida: todo mundo tem uma mãe biológica (por exemplo, nenhum clone) e nenhuma pessoa pode ter duas mães biológicas diferentes.

Pense em funções como ferramentas para descrever relações entre elementos de um conjunto, ou os elementos de dois conjuntos diferentes. Em nossa lista anterior de funções, o Exemplo 2.15 descreve a relação maternal de todas as pessoas (mortas ou vivas), enquanto o Exemplo 2.14 mostra como descrever conjuntos de inteiros utilizando os inteiros.

Pode parecer óbvio que funções dadas por fórmulas são sempre bem definidas, mas podem surgir dificuldades se existir mais de uma maneira de escrever o mesmo elemento do domínio, como mostra o exemplo a seguir.

Exemplo 2.16

Seja $\mathbb{Q} = \left\{ \frac{x}{y} \mid x, y \in \mathbb{Z}, y \neq 0 \right\}$ o conjunto dos números racionais. Ponha

$$f\left(\frac{x}{y}\right) = x + y$$

para todo $\frac{x}{y} \in \mathbb{Q}$. Isso nos dá um função bem definida $f : \mathbb{Q} \to \mathbb{Z}$?

Solução: Não. Como um contraexemplo, note que $\frac{2}{3} = \frac{4}{6}$, mas

$$f\left(\frac{2}{3}\right) = 2 + 3 = 5 \neq f\left(\frac{4}{6}\right) = 4 + 6 = 10,$$

então f não é bem definida.

Exemplo 2.17

Mostre que a função $f : \mathbb{Q} \to \mathbb{Q}$ definida por

$$f\left(\frac{x}{y}\right) = \frac{x+y}{y}$$

está bem definida.

Solução: Seja $\frac{a}{b} = \frac{c}{d} \in \mathbb{Q}$. Então

$$f\left(\frac{a}{b}\right) = \frac{a+b}{b} = \frac{a}{b} + \frac{b}{b} = \frac{c}{d} + \frac{d}{d} = \frac{c+d}{d} = f\left(\frac{c}{d}\right)$$

portanto f está bem definida.

2.2.2 Funções Injetivas e Sobrejetivas

Definição 2.2

Uma função $f : X \to Y$ é *injetiva* se, para todo a e b em X, $f(a) = f(b)$ implica que $a = b$. Nesse caso, nós dizemos que f é uma função injetiva de X para Y.

Definição 2.3

Uma função $f : X \to Y$ é *sobrejetiva* se para todo $y \in Y$ existe um $x \in X$ tal que $f(x) = y$. Nesse caso dizemos que f envia X sobre Y.

Uma função injetiva sempre irá associar elementos diferentes do domínio a diferentes elementos do contradomínio. Em outras palavras, no máximo um elemento do domínio é enviado a um elemento qualquer do contradomínio.

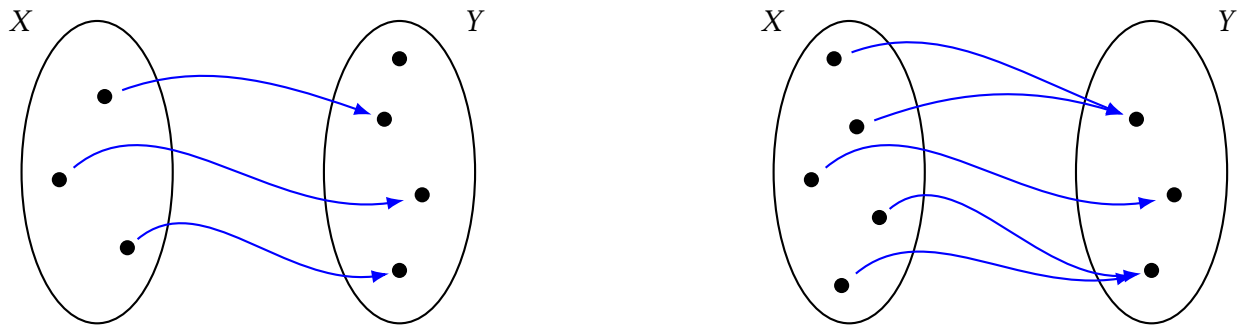


Figura 2.1: Exemplos de uma função injetiva (esquerda) e uma função sobrejetiva (direita).

Usamos o termo *imagem* para descrever o conjunto de todos os valores que uma função pode ter. Uma função sobrejetiva tem cada elemento do contradomínio em sua imagem. A Figura 2.1 ilustra uma função sobrejetiva. Note que o domínio é enviado sobre todo o contradomínio. Lembre como as definições são usadas em matemática: a fim de provar que uma função é injetiva ou sobrejetiva, quase sempre temos que usar a definição.

Exemplo 2.18

Mostre que a função $f : \mathbb{Z} \to \mathbb{Z}$ definida por

$$f(x) = 2x + 1$$

é injetiva.

Solução: Sejam $a, b \in \mathbb{Z}$ e suponha que $f(a) = f(b)$. Então

$$2a + 1 = 2b + 1 \Rightarrow 2a = 2b \Rightarrow a = b$$

Mostramos que $f(a) = f(b)$ implica que $a = b$, ou seja, que f é injetiva.

Definição 2.4: Função Piso

Definimos e denotamos por $\lfloor x \rfloor$ o maior inteiro menor ou igual a x. Então, por exemplo, $\lfloor 4,3 \rfloor = 4$, $\lfloor -2,1 \rfloor = -3$ e $\lfloor 17 \rfloor = 17$. A função que envia x a $\lfloor x \rfloor$ é chamada de função **piso**.

Exemplo 2.19

Seja $f : \mathbb{R} \to \mathbb{Z}$ definida por $f(x) = \lfloor x \rfloor$. Mostre que f envia $\mathbb{R}$ sobre $\mathbb{Z}$.

Solução: Seja $n \in \mathbb{Z}$. Uma vez que $\mathbb{Z} \subseteq \mathbb{R}$, então $n \in \mathbb{R}$ também. Contudo, uma vez que n é um número inteiro, $\lfloor n \rfloor = n$. Portanto, $f(n) = n$ e f é sobrejetiva.

As demonstrações nos dois últimos exemplos são padrões. Para provar que uma função f é injetiva, suponha $f(a) = f(b)$ e mostre que $a = b$. Para mostrar que uma função f é sobrejetiva, considere y um elemento do contradomínio e encontre algum x no domínio tal que $f(x) = y$.

Para provar que uma função não é injetiva ou sobrejetiva, procure um contraexemplo. A função do Exemplo 2.18 não é sobrejetiva porque, por exemplo, $38 \in \mathbb{Z}$, mas não existe um número inteiro x tal que $2x + 1 = 38$. Da mesma forma, a função no Exemplo 2.19 não é injetiva porque $\lfloor 9,3 \rfloor = \lfloor 9,8 \rfloor$, mas $9,3 \neq 9,8$. Note que esses dois exemplos mostram que ser injetiva não é o mesmo que ser sobrejetiva.

É claro que é possível para uma função ser tanto injetiva quanto sobrejetiva. Tal função é chamada de *bijetiva*, ou uma *bijeção*. Compare os exemplos a seguir com o Exemplo 2.18.

Exemplo 2.20

Mostre que a função $f : \mathbb{R} \to \mathbb{R}$ definida por

$$f(x) = 2x + 1$$

é uma bijeção.

Solução: Precisamos mostrar que f é tanto injetiva quanto sobrejetiva.A demonstração de que essa função é injetiva é exatamente a mesma que a do Exemplo 2.18. Para demonstrar que f é sobrejetiva, seja $y \in \mathbb{R}$ qualquer número real. Tome $x = \frac{y-1}{2}$. Então $x \in \mathbb{R}$ e

$$f(x) = f\left(\frac{y-1}{2}\right) = 2\left(\frac{y-1}{2}\right) + 1$$
$$= y - 1 + 1 = y$$

Assim, f é também uma função sobrejetiva.

Exemplo 2.21

Seja $E = \{n \in \mathbb{Z} \mid n \text{ é par}\}$ e seja $O = \{n \in \mathbb{Z} \mid n \text{ é ímpar}\}$ Defina a função $f : E \times O \to \mathbb{Z}$ definida por

$$f(x, y) = x + y.$$

Responda se f é injetiva e/ou sobrejetiva. Prove ou dê contraexemplo.

Solução: Primeiramente mostramos que f não é sobrejetiva. Suponha, ao contrário, que f é sobrejetiva. Uma vez que $2 \in \mathbb{Z}$ é um elemento do contradomínio, existe algum par ordenado $(x, y) \in E \times O$ tal que $f(x, y) = x + y = 2$. Mas, uma vez que x é par e y é ímpar, $x + y$ é ímpar (verifique). Isso contradiz que 2 é par.

A seguir mostramos que f não é injetiva. Note que

$$f(4, -3) = 1 = f(6, -5)$$

mas $(4, -3) \neq (6, -5)$. Este contraexemplo mostra que f não é injetiva.

2.2.3 Composição de funções

Existem algumas maneiras comuns para formar novas funções a partir de antigas. Uma tal construção é a *composição* de duas funções. Se $f : X \to Y$ e $g : Y \to Z$, então $g \circ f$ é uma função de X a Z definida por $(g \circ f)(x) = g(f(x))$.

Exemplo 2.22

Seja $f : \mathbb{R} \to \mathbb{R}$ definida por $f(x) = \lfloor x \rfloor$, e seja $g : \mathbb{R} \to \mathbb{R}$ definida por $g(x) = 3x$. Então

$$(g \circ f)(2,4) = g(f(2,4)) = g(2) = 6 \qquad \text{e} \qquad (f \circ g)(2,4) = f(g(2,4)) = f(7,2) = 7$$

Esse exemplo mostra que $f \circ g$ pode ser diferente de $g \circ f$. Perceba uma coisa potencialmente confusa sobre a notação: na composição $g \circ f$ fazemos f primeiro, e depois aplicamos g ao resultado. A ordem importa, em geral.

Algumas vezes gostaríamos de ser capazes de "desfazer" uma função, de forma que ela envie cada ponto de volta para onde ele veio. Se $f : X \to Y$ é uma função, então a *função inversa* de f é a função

$$f^{-1} : Y \to X$$

que tem a propriedade que $f^{-1} \circ f = \mathbf{id}_X$ e $f \circ f^{-1} = \mathbf{id}_Y$, onde $\mathbf{id}_A$ denota a chamada *função identidade* que associa cada elemento do conjunto A a si próprio.

Nem todas as funções têm inversas. Se $f : X \to Y$ tem uma inversa, então, para qualquer $y \in Y$, $f(f^{-1}(y)) = y$, de maneira que f deve enviar X sobre Y por completo. Além disso, se $f(a) = f(b)$, então podemos aplicar f^{-1} em ambos os lados dessa equação para ter $a = b$, de modo que f também deve ser injetiva. Assim, vemos que, se uma função possui uma inversa, ela deve ser uma bijeção.

De forma recíproca, podemos construir a inversa de qualquer bijeção $f : X \to Y$ ao pegar $f^{-1}(y)$ como o único elemento de X que é enviado em y. Sabemos que tal elemento existe, porque f é sobrejetiva, e sabemos que esse elemento é único, uma vez que f é injetiva.

Exemplo 2.23

Se $f : \mathbb{R} \to \{y \in \mathbb{R} \mid y > 0\}$ é a função $f(x) = 2^x$, então a inversa de f é dada por $f^{-1}(x) = \log_2 x$.

Nesse último exemplo, poderíamos definir $f(x) = 2^x$ como uma função de $\mathbb{R}$ em $\mathbb{R}$, mas então ela não seria invertível porque não seria sobrejetiva.

2.2.4 Exercícios Propostos

1 Escreva as definições de função injetiva e de sobrejetiva usando termos da lógica de predicados.

2 Seja P um conjunto de pessoas, e seja Q um conjunto de ocupações. Determine uma função $f : P \to Q$ definindo $f(p)$ igual à ocupação de p. O que deve ser verdade a respeito das pessoas em P para que f seja uma função bem definida?

3 Defina um função $t : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \times \mathbb{R}$ por $t(a,b) = (a+b, a-b)$. Prove que t é uma bijeção.

4 Considere a função negação $n : \{V,F\} \to \{V,F\}$ dada por $n(x) = \neg x$. Essa função é uma bijeção? O que é n^{-1}?

5 Defina uma função $f : \mathbb{Z} \to \mathbb{Z} \times \mathbb{Z}$ por $f(x) = (2x+3, x-4)$.

a) f é injetiva? Prove ou dê um contraexemplo.

b) f é sobrejetiva? Mostre porque ou refute.

6 Seja $f : \mathbb{Z} \to \mathbb{Z}$ definida por

$$f(x) = \begin{cases} x+3 & \text{se } x \text{ é ímpar} \\ x-5 & \text{se } x \text{ é par.} \end{cases}$$

a) Mostre que f é uma bijeção.

b) Encontre uma fórmula para f^{-1}.

7 Calcule a inversa de $f(x) = x^3 + 1$. Verifique sua resposta mostrando que $f(f^{-1}(x)) = x$ e $f^{-1}(f(x)) = x$.

8 Similar ao conceito da função piso, podemos definir a função *teto* denotando $\lceil x \rceil$ como sendo o menor inteiro maior ou igual a x. Encontre os valores abaixo para as funções piso e teto:

a) $\lceil 1, 1 \rceil$
b) $\lceil -\frac{3}{4} \rceil$
c) $\lfloor \frac{7}{8} \rfloor$
d) $\lfloor -2, 99 \rfloor$
e) $\lceil 3 \rceil$
f) $\lfloor \frac{1}{2} + \lceil \frac{3}{2} \rceil \rfloor$
g) $\lfloor \frac{1}{2} \cdot \lfloor \frac{5}{2} \rfloor \rfloor$
h) $\lceil \lfloor \frac{1}{2} \rfloor + \lceil \frac{1}{2} \rceil + \frac{1}{2} \rceil$

2.3 Algumas funções importantes em Computação

Listamos aqui algumas funções que costumam aparecer em algoritmos e linguagens de programação.

2.3.1 Quociente inteiro e resto

Os conceitos de divisão (quociente) e resto de um número natural x por um inteiro positivo d são conhecidos e consensuais desde a antiguidade: 17 dividido por 3 é 5 com resto 2. O resto dessa divisão é também chamado "x módulo d".

Em matemática, a divisão inteira é indicada às vezes pelo símbolo antigo '÷', e o resto pela sigla 'mod', ambos usados como operações entre dois inteiros. Dessa forma podemos escrever $17 \div 3 = 5$ e $17 \bmod 3 = 2$.

Estas operações podem ser definidas usando a função piso:

$$x \div d = \left\lfloor \frac{x}{d} \right\rfloor$$

$$x \bmod d = x - d(x \div d) = x - d \left\lfloor \frac{x}{d} \right\rfloor$$

Em matemática, estas fórmulas são adotadas como definições dessas duas operações também no caso de x ser um inteiro negativo. Assim, $(-17) \div 3 = \lfloor \frac{17}{3} \rfloor = -6$, e portanto $(-17) \bmod 3 = (-17) - 3(-6) = 1$.

Mais geralmente, estas fórmulas também são usadas quando x é um número real, inclusive negativo, e d é um número real positivo. Assim, por exemplo, $3.1416 \div 0.001 = 3141$, e $3.1416 \bmod 0.001 = 0.0006$.

Algumas linguagens de programação modernas, como Python, usam as definições acima, embora com outros símbolos. Outras linguagens, como C e Fortran, calculam $\|x\| \div d$ e $\|x\| \bmod d$, e devolvem o resultado com o sinal de x. Nessas linguagens, por exemplo, $(-17) \div 3 = -5$ (em vez de -6).

e $(-17) \bmod 3 = -2$ (em vez de 1). Embora pareçam menos naturais, as definições matemáticas são geralmente mais úteis, em matemática e programação, do que estas definições antigas.

Não há consenso sobre a definição de $x \div d$ ou $x \bmod d$ quando d é negativo. Felizmente, este caso raramente ocorre, na prática ou na teoria.

Diz-se que dois números x e y são congruentes módulo d se $(x \bmod d) = (y \bmod d)$, ou seja se $(x - y) \bmod d = 0$.

Exemplo 2.24

Os militares estadunidenses marcam o tempo usando 24 horas, com as horas variando de 0 a 23. Meia hora antes de meia-noite para os militares é 23:30. Os civis estadunidenses costumam marcar o tempo usando 12 horas; diferenciam a manhã e a tarde usando AM (do latim *ante meridiem*, antes do meio-dia) para a manhã e PM (*post meridiem*) para a tarde. Meia hora antes de meia-noite para os civis é 11:30 PM. A conversão do horário militar para o horário civil usa uma função módulo 12:

8:00 no horário militar: Calcule 8 mod 12 = 8, obtendo 8:00 AM no horário civil

16:00 no horário militar: Calcule 16 mod 12 = 4, obtendo 4:00 PM no horário civil

O sufixo AM ou PM é determinado pelo quociente na divisão, que pode ser 0 (AM) ou 1 (PM). A contagem módulo 12 começa em 0 e recomeça quando se atinge 12:

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2, 3, \ldots$$

Esse sistema de contagem é ensinado para crianças pequenas algumas vezes como "aritmética do relógio".

2.3.2 Dispersão

Uma **função de dispersão** é uma função $h : S \to T$ tal que o domínio S é um conjunto de cadeias de texto ou valores inteiros e o contradomínio T é o conjunto de inteiros $\{0, 1, \ldots, t-1\}$, em que t é algum inteiro positivo relativamente pequeno. Se o domínio S consistir em cadeias de textos, podemos imaginá-las codificadas de alguma forma como valores inteiros, talvez por um algoritmo tão simples como o que converte cada letra individual em uma cadeia de texto em sua posição no alfabeto ($a = 1$, $b = 2$ e assim por diante) e depois soma a lista de inteiros resultante para obter um valor inteiro. Podemos, então, começar supondo que S consiste em valores inteiros.

A função h, portanto, leva um conjunto possivelmente grande de inteiros S em uma janela relativamente pequena de valores inteiros T. Logo, h não deve ser uma função injetora, já que podem existir muitos valores $x_1, x_2, x_3, \ldots$ em S tais que $h(x_1) = h(x_2) = h(x_3)$; nesse caso, dizemos que o valor da função foi "dispersado" para x_1, x_2, x_3. Seja o que for que a função $h(x)$ faça com x, o último passo é quase sempre aplicar a função $\bmod t$ para que o resultado final caia no contradomínio $\{0, 1, \ldots, t-1\}$.

Exemplo 2.25

Sejam S um conjunto de inteiros positivos, T o conjunto de inteiros $\{0, 1, \ldots, t-1\}$ e a função de dispersão h é dada por

$$h(x) = x \bmod t$$

Se $t = 10$, por exemplo, então os valores são calculados módulo 10:

$$h(7) = 7 \bmod 10 = 7$$
$$h(23) = 23 \bmod 10 = 3$$
$$h(59) = 59 \bmod 10 = 9$$
$$h(158) = 158 \bmod 10 = 8$$
$$h(48) = 48 \bmod 10 = 8$$

Aqui o valor 8 foi dispersado para 158 e 48.

Uma função de dispersão é usada com frequência em algoritmos de busca. Na busca sequencial, n elementos estão armazenados em uma lista não ordenada (aleatória) e um valor alvo dado é comparado com todos os elementos da lista, um por um. Na busca binária, n elementos estão armazenados em uma lista ordenada. O valor alvo dado é comparado com o ponto do meio da lista, depois com o ponto do meio de uma das metades da lista e assim por diante. Em uma busca usando uma função de dispersão, n elementos estão armazenados em um array (uma tabela unidimensional) chamado de tabela de dispersão, com o array indexado de 0 a $t - 1$; a tabela tem tamanho t. O elemento x é passado como argumento da função de dispersão, e o valor $h(x)$ resultante fornece o índice no array onde o elemento é armazenado. Mais tarde, quando é feita uma busca, o valor alvo é passado pela mesma função de dispersão, fornecendo um índice para o local na tabela de dispersão, onde o elemento alvo será procurado.

Exemplo 2.26

Registros de votação para os cidadãos em determinado distrito dos EUA são armazenados em uma tabela de dispersão de tamanho 197 usando o número da Previdência Social com um valor-chave (o valor que será usado pela função de dispersão). Usando uma função de dispersão módulo o tamanho da tabela de dispersão, o índice para se buscar primeiro pelos dados sobre um eleitor com número da Previdência Social

$$328356770$$

é

$$328356770 \bmod 197 = 125$$

Funções de dispersão também são conhecidas como funções **HASH** e são um componente fundamental de Segurança computacional (ou o termo mais geral, segurança da informação), um tópico de interesse crítico, já que nossa economia, nossos interesses e, de fato, toda a nossa vida dependem tanto de computadores e de informação. A função mod aparece em muitos aspectos de segurança, em especial em **Criptografia**, o estudo dos vários esquemas de codificação/decodificação. O termo mais amplo, criptologia, inclui não só a criptografia, mas também as técnicas usadas para analisar e "quebrar" os códigos de mensagens cifradas. Um dos exemplos mais famosos de códigos que foram quebrados ocorreu durante a Segunda Guerra Mundial, quando um time de analistas ingleses, incluindo o matemático Alan Turing, trabalhando em Bletchley Park, conseguiu quebrar o código "Enigma", supostamente invencível, e decifrar os planos dos movimentos dos submarinos alemães.

Capítulo 3

Pensamento Quantitativo

Capítulo 4

Pensamento Relacional