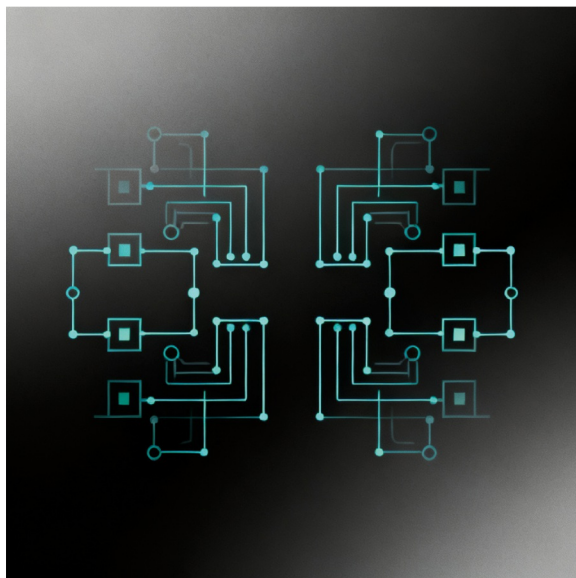


Lógica Formal

Lógica proposicional, lógica de predicados, lógica modal e lógicas não clássicas



Lucien Sina

Lógica Formal

Lógica proposicional, lógica de predicados,
lógica modal e lógicas não clássicas

Lucien Sina

13.7.2026

Sumário

0.1	Introdução e visão geral	3
0.1.1	Motivação e relevância	3
0.1.2	Contexto histórico	4
0.1.3	Objetivos deste livro	4
0.1.4	Procedimento e estrutura	5
0.1.5	A quem se destina este livro? . . .	6
0.1.6	Objetivos de aprendizagem e orientações de uso	6
0.1.7	Breves descrições dos capítulos . .	8
0.2	Fundamentos	11
0.2.1	Notação e teoria dos conjuntos . .	11
0.2.2	Estruturas e semântica	14
0.2.3	Princípios de indução	14
0.3	Estrutura e estilo de textos matemáticos .	19
1	Lógica proposicional	22
1.1	Introdução e visão geral	22
1.2	Sintaxe e semântica	25
1.2.1	Introdução	25
1.2.2	Assinaturas	25

1.2.3	Fórmulas	26
1.2.4	Semântica: interpretações, satisfação e modelos	27
1.2.5	Tabelas-verdade	30
1.2.6	Funções de verdade	30
1.3	Exercícios com soluções	34
1.4	Inferência	39
1.4.1	Modelos como descrição de situações	40
1.4.2	Operações sobre conjuntos de modelos	40
1.4.3	Classes de satisfatibilidade de fórmulas	44
1.4.4	Consequência semântica	47
1.4.5	Fechamento dedutivo (O que decorre das premissas?)	48
1.4.6	Teorias e a relação Th/Mod	51
1.4.7	Independência em relação a átomos supérfluos	54
1.4.8	Breve lista de verificação para argumentos semânticos	61
1.4.9	Exercício (curto)	61
1.5	Exercícios e soluções: inferência	62
1.6	Equivalência	66
1.6.1	Fundamentos	66
1.6.2	Equivalências específicas	71
1.6.3	Formas normais	80
1.6.4	Exercícios e soluções — equivalência	90
1.7	Consideração algorítmica	95
1.7.1	Problemas de inferência	95

1.7.2	Cálculos — procedimentos sintáticos de inferência	100
1.7.3	Cálculo de resolução	105
1.7.4	Exercícios e soluções	118
1.8	Solucionadores SAT	120
1.8.1	Equivalência quanto à satisfatibilidade	121
1.8.2	A codificação de Tseitin	125
1.8.3	O algoritmo DPLL	134
1.8.4	Exercícios e soluções	140
2	Lógica de predicados	143
2.1	Introdução e visão geral	143
2.2	Sintaxe e semântica	146
2.2.1	Assinaturas	148
2.2.2	Termos e variáveis	152
2.2.3	Fórmulas	155
2.2.4	Semântica	160
2.2.5	Exercícios e soluções	167
2.3	Coincidência, equivalência e substituição	170
2.3.1	Lema da coincidência	171
2.3.2	Equivalência	175
2.3.3	Substituição	193
2.3.4	Classes de modelos	208
2.3.5	Exercícios e soluções	212
2.4	Formas normais e resolução básica	214
2.4.1	Semântica para conjuntos de fórmulas	215
2.4.2	Formas normais	218

2.4.3	Modelos de Herbrand	231
2.4.4	Teorema de Herbrand	238
2.4.5	Resolução básica	246
2.4.6	Exercícios e soluções	252
2.5	Consequências e resolução	253
2.5.1	Consequência lógica	254
2.5.2	Unificação	258
2.5.3	Resolução na lógica de predicados	267
2.5.4	Teorema da completude de Gödel .	278
2.5.5	Fechos dedutivos e teorias	283
2.5.6	Lema de Lindenbaum: existência de teorias maximais consistentes .	299
2.5.7	Construção de Henkin	302
2.5.8	Teorema de Henkin	305
2.5.9	O teorema da completude de Gödel	309
2.5.10	Do teorema da completude aos teoremas da incompletude	312
2.5.11	Os teoremas da incompletude de Gödel	327
2.5.12	Computabilidade e complexidade .	337
2.5.13	Exercícios e soluções	342
2.6	Lógica descritiva	350
2.6.1	Conceitos, papéis e sua interpretação	351
2.7	Lógica de predicados de segunda ordem .	389
2.7.1	Sintaxe e semântica	390
2.7.2	Propriedades e expressividade . . .	395
2.7.3	Lógica e complexidade descritiva .	411
2.7.4	Exercícios e soluções	415

3	Lógica modal	419
3.1	Lógica modal geral	422
3.1.1	Sintaxe e semântica	423
3.1.2	Conceitos e propriedades	429
3.1.3	Consequência lógica e cálculo de tableaux	438
3.1.4	Exercícios e soluções	448
3.2	Lógica temporal linear	456
3.2.1	Sintaxe e semântica	457
3.2.2	Equivalências e formas normais . .	462
3.2.3	Relação com estruturas de Kripke	474
3.2.4	Distinção de estados em estruturas de Kripke	480
3.2.5	Relação com a lógica de predicados	488
3.2.6	Exercícios e soluções	494
3.3	Lógica temporal ramificada	497
3.3.1	Sintaxe e semântica	498
3.3.2	Equivalências e formas normais . .	506
3.3.3	Bissimulação	519
3.3.4	Relação com LTL	529
3.3.5	Exercícios e soluções	535
3.4	Lógica epistêmica	541
3.4.1	Restrições sobre quadros	542
3.4.2	Lógica modal para múltiplos agentes	552
3.4.3	Anúncios públicos	558
3.4.4	Exercícios e soluções	564
4	Lógicas não clássicas	569
4.1	Lógica intuicionista	572

4.1.1	Semântica de Kripke	578
4.1.2	Propriedades	586
4.1.3	Semântica baseada em provas . . .	596
4.1.4	Exercícios com soluções	615
4.2	Lógica paraconsistente	617
4.2.1	Lógica proposicional trivalorada .	618
4.2.2	Inferência com modelos minimais .	630
4.2.3	Lógica proposicional tetravalorada	641
4.2.4	Exercícios e soluções	656
4.3	Lógica fuzzy	659
4.3.1	Conjuntos fuzzy	660
4.3.2	T-normas	667
4.3.3	S-normas	674
4.3.4	Negações	682
4.3.5	Semântica e inferências	692
4.3.6	Exercícios com soluções	702
4.4	Lógica default	705
4.4.1	Conclusões não monotônicas . . .	706
4.4.2	Sintaxe e semântica da lógica de- fault de Reiter	710
4.4.3	Inferência e semimonotonicidade .	724
4.4.4	Exercícios e soluções	731
4.5	Programação em conjuntos de respostas .	735
4.5.1	Programas lógicos estendidos . . .	736
4.5.2	Modelos minimais	741
4.5.3	Conjuntos de respostas	753
4.5.4	Exercícios e soluções	763

5	Considerações finais	771
5.1	Resumo	771
5.2	Aprofundamentos e aplicações	774
5.3	Recomendações de leitura	779
6	Aviso legal	788

Prefácio

A lógica formal é um dos fundamentos do pensamento científico. Ela oferece uma linguagem precisa para analisar proposições, conceitos, inferências e demonstrações, criando assim ferramentas de importância central para a matemática, a informática, a filosofia, a linguística e muitas outras disciplinas. Quem compreende as relações lógicas adquire não apenas uma compreensão mais profunda dos sistemas formais, mas também da estrutura dos argumentos e dos próprios limites da inferência.

Este livro pretende oferecer uma introdução compreensível e, ao mesmo tempo, sistemática à lógica formal. Ele se dirige a leitoras e leitores que desejam estudar procedimentos lógicos de maneira básica ou aprofundada, acompanhando-os desde os primeiros conceitos da lógica proposicional, passando pela lógica de predicados, até as lógicas modais e algumas lógicas não clássicas selecionadas. Dá-se especial importância a introduzir os temas não apenas de modo formal, mas também a torná-los visíveis em suas relações mútuas.

No centro deste livro não está apenas o aprendizado de definições e teoremas, mas o desenvolvimento de uma compreensão sólida dos métodos lógicos. Por isso, conceitos centrais como sintaxe, semântica, modelo, consequência, equivalência e forma normal são construídos cuidadosamente e complementados por exemplos, exercícios e soluções. O percurso vai dos fundamentos da inferência formal, passando por procedimentos algorítmicos como resolução e SAT-solvers, até temas mais avançados como teoria de Herbrand, lógica descritiva, lógica epistêmica, lógica fuzzy, lógica default e programação em conjuntos de respostas.

O livro segue uma abordagem didática que combina precisão formal com desenvolvimento gradual. Muitas seções são elaboradas de modo a apresentar não apenas resultados isolados, mas também sua motivação, seu significado e suas aplicações. Assim, a lógica não deve aparecer como uma coleção de regras isoladas, mas como uma área coerente, dotada de uma estrutura interna clara e de grande alcance prático.

Meus agradecimentos vão a todas as pessoas que contribuíram para a elaboração deste livro com sugestões, comentários e observações críticas. Agradeço igualmente às leitoras e aos leitores que se dispõem a empreender esta jornada pela lógica formal, por vezes exigente, mas recompensadora. Comentários, correções e propostas de melhoria são sempre bem-vindos, pois ajudam a tornar o livro mais preciso e melhor.

Desejo uma leitura proveitosa e estimulante.

0.1 Introdução e visão geral

A lógica é a disciplina que investiga sistematicamente o raciocínio inferencial. No centro estão as linguagens formais (sintaxe), que estabelecem as regras para a formação de expressões bem formadas, e as semânticas, que atribuem significado a essa sintaxe. Com essas ferramentas, questões relativas à verdade, à consequência lógica e à representação do conhecimento podem ser formuladas e analisadas com precisão. Uma lógica fornece, assim, tanto uma descrição de quais expressões são válidas quanto um modelo de como as proposições devem ser interpretadas em relação a estruturas – os chamados modelos.

0.1.1 Motivação e relevância

O estudo da lógica é central por várias razões. Por um lado, a lógica constitui a base formal da matemática e de muitas áreas da informática. Conceitos como algoritmo, computabilidade e complexidade estão historicamente estreitamente ligados à lógica formal; sua formação conceitual e suas técnicas de demonstração são fundamentais para a compreensão daquilo que pode ser resolvido por meios automáticos. Por outro lado, a lógica possui aplicações diretas: especificação e verificação de software e hardware, representação do conhecimento

na inteligência artificial, consultas em bancos de dados, raciocínio automático em provadores de teoremas, bem como modelagem nas formalizações de muitas ciências.

0.1.2 Contexto histórico

A lógica formal moderna tem raízes na filosofia e na matemática dos séculos XIX e XX. Etapas notáveis desse desenvolvimento incluem Frege, Russell e Hilbert, que impulsionaram os sistemas formais e as apresentações axiomáticas, bem como Turing, Church e Gödel, cujos resultados esclareceram a relação entre computabilidade e demonstrabilidade formal. Desses desenvolvimentos surgiram conceitos centrais que ainda hoje constituem o quadro de referência para o tratamento técnico e teórico da lógica.

0.1.3 Objetivos deste livro

Este livro persegue três objetivos complementares:

1. Oferecer uma introdução precisa e autocontida às lógicas mais importantes (lógica proposicional, lógica de predicados, lógicas modais, lógicas não clássicas selecionadas e programação em conjuntos de respostas).
2. Estabelecer a ponte com a informática: são tratados aspectos algorítmicos, questões de decidibilidade e complexidade, bem como procedimentos concre-

tos (por exemplo, SAT-solvers, model checking e programação em conjuntos de respostas).

3. Desenvolver competências metodológicas: formular corretamente, demonstrar por indução (completa e estrutural), bem como lidar com sintaxe e semântica.

0.1.4 Procedimento e estrutura

Neste livro seguimos o seguinte padrão recorrente:

- **Sintaxe:** definimos a linguagem formal por meio de assinatura, termos e fórmulas; apresentamos regras de formação e notações.
- **Semântica:** descrevemos como as fórmulas são interpretadas em estruturas e definimos noções de verdade, satisfatibilidade e validade.
- **Procedimentos de prova e decisão:** para cada lógica, consideramos sistemas usuais de prova (axiomáticos, dedução natural, resolução ou similares), bem como procedimentos algorítmicos e seus limites.
- **Exemplos e aplicações:** ao final de cada capítulo central, tratamos aplicações típicas e propomos exercícios.

0.1.5 A quem se destina este livro?

O livro destina-se a estudantes avançados de graduação e mestrado em informática, matemática e áreas afins, bem como a profissionais que buscam uma compreensão sólida e matematicamente fundamentada da lógica. Pressupõem-se conhecimentos básicos de teoria elementar dos conjuntos, matemática discreta e técnicas de demonstração (em especial indução completa); quando necessário, retomamos brevemente esses fundamentos.

O que este livro não pretende oferecer

Esta obra não é uma coletânea abrangente de exercícios de programação para ferramentas lógicas específicas, nem uma exposição histórica completa de todas as escolas da lógica. Em vez disso, concentra-se nos conceitos formais, em algoritmos típicos e na relação com a prática da informática. Aspectos especializados (por exemplo, a implementação detalhada de SMT-solvers ou SAT-solvers) são tratados apenas na medida em que servem para ilustrar afirmações teóricas.

0.1.6 Objetivos de aprendizagem e orientações de uso

Após a leitura dos seis primeiros capítulos, leitoras e leitores deverão ser capazes de:

- definir linguagens formais e descrever sua semântica em estruturas;

- aplicar técnicas típicas de demonstração da lógica (indução completa e estrutural, prova por contradição, construção direta);
- compreender em seus traços fundamentais procedimentos algorítmicos como resolução SAT e model checking, bem como executar manualmente casos simples;
- reconhecer os limites dos sistemas formais (indecidibilidade, classificação de complexidade) e deles extrair consequências para a informática.

Observações sobre a notação

Procuramos manter uma notação consistente; as convenções centrais são:

- $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ denotam os conjuntos numéricos usuais;
- variáveis proposicionais ou de fórmulas são geralmente denotadas por p, q, r, φ, ψ , e indivíduos por x, y, z ;
- para operações com conjuntos, usamos $\cup, \cap, \setminus, \times$;
- quantificadores aparecem na forma $\forall x$ ou $\exists x$.

Próximos passos

Em seguida, apresentamos uma introdução concisa, mas completa, à notação e retomamos a teoria dos conjuntos e

a teoria da indução necessárias. Depois disso, começamos com a lógica proposicional e avançamos gradualmente para os temas mais avançados.

Na sequência, encontram-se breves resumos de cada capítulo, que oferecem uma orientação sobre os conteúdos e as competências desenvolvidos nas respectivas seções.

0.1.7 Breves descrições dos capítulos

Nos parágrafos seguintes, oferecemos uma orientação sucinta sobre os temas tratados em cada capítulo e sobre as competências que a leitora ou o leitor deverá adquirir a partir deles.

Capítulo 0 – Introdução e fundamentos. Este capítulo estabelece os fundamentos notacionais e conceituais centrais: teoria dos conjuntos e das relações, funções, princípios de indução, construção de linguagens formais, bem como noções básicas de semântica (modelos, satisfatibilidade, validade, consequência lógica). Ele serve como referência à qual se recorrerá repetidamente nos capítulos seguintes.

Capítulo 1 – Lógica proposicional. A lógica proposicional constitui a porta de entrada para o raciocínio formal: introduzimos variáveis proposicionais e conectivos (“e”, “ou”, “não”, “implica”), definimos a avaliação semântica por meio de tabelas de verdade e tratamos formas normais sintáticas (FNC, FND). Temas importantes são

procedimentos de prova (procedimentos baseados em resolução, dedução natural), satisfatibilidade (SAT) e afirmações fundamentais sobre complexidade.

Capítulo 2 – Lógica de predicados. A lógica de predicados amplia a lógica proposicional por meio de quantificadores e predicados, de modo que relações entre indivíduos possam ser expressas. Tratamos assinaturas, estruturas, interpretação de termos e fórmulas, bem como resultados metateóricos centrais (completude, incompletude, Löwenheim–Skolem, questões de decidibilidade) e mostramos suas consequências para a informática (por exemplo, limites da automatização).

Capítulo 3 – Lógicas modais (incl. lógica temporal). As lógicas modais permitem qualificar enunciados por meio de operadores como “possível”, “necessário”, “sempre” e “eventualmente”. Introduzimos a semântica de Kripke, tratamos sistemas modais típicos (K, T, S4, S5) e descrevemos instâncias especiais, como as lógicas temporais (LTL, CTL), bem como suas aplicações em model checking e na especificação de sistemas.

Capítulo 4 – Outras lógicas não clássicas e programação em conjuntos de respostas. Aqui consideramos uma seleção de lógicas não clássicas que representam extensões típicas ou alternativas à lógica clássica: lógica intuicionista, lógica paraconsistente, lógicas multivaloradas e lógica fuzzy, lógicas default e probabilísticas.

Além disso, apresentamos a programação em conjuntos de respostas (Answer Set Programming, ASP) como uma técnica prática de representação do conhecimento e de inferência não monotônica.

Capítulo 5 – Síntese e perspectivas. O capítulo final resume resultados centrais, mostra conexões entre as áreas tratadas e oferece uma breve perspectiva sobre raciocínio automático, temas atuais de pesquisa e aplicações avançadas na informática, na inteligência artificial e na matemática.

Conceitos importantes que reaparecem ao longo deste livro são:

- **Fórmula** – uma expressão sintaticamente bem formada de uma linguagem lógica.
- **Estrutura ou modelo** – uma interpretação que atribui valores aos símbolos da linguagem.
- **Satisfatibilidade** – existência de um modelo que torna uma fórmula verdadeira.
- **Consequência** – relação na qual, a partir de um conjunto de fórmulas, outra fórmula segue logicamente.
- **Model checking** – verificação algorítmica de se uma estrutura é modelo de uma fórmula.

Este livro tem por objetivo oferecer uma introdução precisa e orientada a aplicações às lógicas centrais. Valorizamos a precisão formal, exemplos motivadores e aspectos algorítmicos como computabilidade e complexidade.

0.2 Fundamentos

0.2.1 Notação e teoria dos conjuntos

Usamos a notação usual da teoria dos conjuntos. Se X é um conjunto e x é um objeto, então $x \in X$ significa que x é um elemento de X . Para dois conjuntos X e Y , escrevemos $X \subseteq Y$ quando X é um subconjunto de Y .

Os conjuntos dos números naturais, dos números inteiros e dos números reais são denotados por $\mathbb{N}$, $\mathbb{Z}$ e $\mathbb{R}$.

O produto cartesiano de dois conjuntos X e Y é denotado por $X \times Y$. A potência de um conjunto X é escrita como $\mathcal{P}(X)$ ou 2^X .

Conjuntos, operações e relações

Conjuntos podem ser representados implicitamente por meio de condições de construção. Por exemplo,

$$D = \left\{ x \in \mathbb{N} \left| \begin{array}{l} \exists y \in \mathbb{N} : y \mid x \\ \text{e } 1 < y < x \end{array} \right. \right\}$$

é o conjunto de todos os números naturais compostos.

Operações importantes com conjuntos são a união $\cup$, a interseção $\cap$ e a diferença $\setminus$. Para dois conjuntos X, X' , vale:

$$X \cup X' = \{x \mid x \in X \text{ ou } x \in X'\},$$

$$X \cap X' = \{x \mid x \in X \text{ e } x \in X'\},$$

$$X \setminus X' = \{x \mid x \in X \text{ e } x \notin X'\}.$$

O produto cartesiano de X e X' é

$$X \times X' = \{(x, x') \mid x \in X, x' \in X'\}.$$

Mais geralmente, escrevemos $X^n = X \times \cdots \times X$ (n vezes) para o produto cartesiano n -ário.

Uma relação (binária) R entre os conjuntos X e X' é um subconjunto $R \subseteq X \times X'$. Em vez de $(x, x') \in R$, escrevemos ocasionalmente também xRx' . Uma relação n -ária sobre X é um subconjunto de X^n .

Se $R \subseteq X \times X$ (isto é, $X' = X$), definimos:

- R é *reflexiva* se, e somente se, $(x, x) \in R$ vale para todo $x \in X$.
- R é *simétrica* se, e somente se, de $(x, y) \in R$ também se segue $(y, x) \in R$.
- R é *transitiva* se, e somente se, de $(x, y) \in R$ e $(y, z) \in R$ também se segue $(x, z) \in R$.

Se R é reflexiva, simétrica e transitiva, então R é chamada de *relação de equivalência*.

Uma função f é uma aplicação $f : X \rightarrow Y$. Para $x \in X$, denotamos por $f(x)$ a imagem de x . Toda função

também pode ser entendida como uma relação especial $f \subseteq X \times Y$ que satisfaz a propriedade de funcionalidade: de $(x, y) \in f$ segue-se que não existe $y' \in Y$ com $y' \neq y$ e $(x, y') \in f$.

Seja $f : X \rightarrow Y$.

- f é *injetiva* (uma aplicação um-para-um), sse, para todos $x, x' \in X$, de $f(x) = f(x')$ sempre se segue $x = x'$.
- f é *sobrejetiva*, sse, para todo $y \in Y$, existe um $x \in X$ com $f(x) = y$.
- f é *bijetiva*, sse é injetiva e sobrejetiva.

A aplicação inversa f^{-1} é definida, em geral, como uma aplicação para o conjunto das partes:

$$f^{-1} : Y \rightarrow \mathcal{P}(X), \quad f^{-1}(y) = \{x \in X \mid f(x) = y\}.$$

Se f é uma bijeção, então f^{-1} é uma aplicação $f^{-1} : Y \rightarrow X$ e, para cada $y \in Y$, $f^{-1}(y)$ é o único $x \in X$ tal que $f(x) = y$.

Se $f : X \rightarrow Y$ e $g : Y \rightarrow Z$ são aplicações, então $g \circ f : X \rightarrow Z$ denota a composição, definida por $(g \circ f)(x) = g(f(x))$ para todo $x \in X$.

Notação de crescimento. Para funções $f, g : \mathbb{N} \rightarrow \mathbb{N}$, escreve-se $g \in O(f)$, sse existem constantes $c > 0$ e $n_0 \in \mathbb{N}$ tais que

$$\forall n \geq n_0 : \quad g(n) \leq c \cdot f(n).$$

Intuitivamente, $g \in O(f)$ significa que g não cresce assintoticamente mais rápido do que f .

0.2.2 Estruturas e semântica

Uma estrutura $\mathcal{A}$ para uma assinatura lógica consiste em um conjunto universo A e em uma interpretação dos símbolos da assinatura. Uma fórmula φ é verdadeira em uma estrutura $\mathcal{A}$ quando vale $\mathcal{A} \models \varphi$.

Conceitos importantes são:

- Satisfatibilidade: existe uma estrutura $\mathcal{A}$ tal que $\mathcal{A} \models \varphi$
- Validade: para todas as estruturas $\mathcal{A}$, vale $\mathcal{A} \models \varphi$
- Consequência lógica: $\Phi \models \psi$

0.2.3 Princípios de indução

Provas por indução são técnicas de demonstração centrais na lógica e na informática teórica. Distinguimos aqui dois tipos usuais: a *indução completa* sobre os números naturais e a *indução estrutural* sobre objetos definidos sintaticamente (termos, fórmulas, árvores). Ambos os procedimentos se baseiam no princípio de estruturas bem ordenadas ou bem fundadas, mas diferem na formulação do passo indutivo.

Indução completa

A indução completa (também chamada de indução simples ou forte, conforme a formulação) aplica-se a afirmações sobre números naturais. Para provar uma afirmação $A(n)$ para todo $n \in \mathbb{N}$, mostra-se:

1. (*base da indução*) $A(1)$ vale (ou $A(0)$, conforme a convenção).
2. (*passo indutivo*) Para um $n > 1$ arbitrário, segue da hipótese de que $A(k)$ vale para todo $k < n$ que também $A(n)$ vale.

A indução completa decorre então do fato de que não pode existir um conjunto de contraexemplos com menor elemento.

Exemplo 0.1. (*Fórmula da soma de Gauss*) Para todo $n \in \mathbb{N}$, vale

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}.$$

Demonstração. Base da indução: Para $n = 1$, temos $\sum_{i=1}^1 i = 1$ e $\frac{1 \cdot (1+1)}{2} = 1$. Portanto, a igualdade é verdadeira para $n = 1$.

Passo de indução: Seja $n > 1$ e suponhamos que a fórmula valha para $n - 1$, isto é,

$$\sum_{i=1}^{n-1} i = \frac{(n-1)n}{2}.$$

Então, para n , obtemos:

$$\begin{aligned}
 \sum_{i=1}^n i &= \left(\sum_{i=1}^{n-1} i \right) + n \\
 &= \frac{(n-1)n}{2} + n \\
 &= \frac{n(n-1) + 2n}{2} \\
 &= \frac{n(n+1)}{2}.
 \end{aligned}$$

Com isso, o passo indutivo está demonstrado e a afirmação fica provada para todo n . $\square$

Indução estrutural

A indução estrutural é adequada para afirmações sobre objetos definidos recursivamente e também sobre objetos com estrutura de árvore, como termos ou fórmulas. Toma-se a definição recursiva do tipo de objeto como guia: mostra-se a propriedade para os objetos elementares (atômicos) e, em seguida, que a propriedade se preserva ao compor objetos segundo as regras de construção.

Formalmente: seja $\mathcal{F}$ o conjunto das fórmulas de uma linguagem, definido pelas regras

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi)$$

com fórmulas atômicas p . Para mostrar uma propriedade $P(\varphi)$ para todo $\varphi \in \mathcal{F}$, prova-se:

- $P(p)$ para cada fórmula atômica p .
- De $P(\varphi)$ segue $P(\neg\varphi)$.
- De $P(\varphi)$ e $P(\psi)$ segue $P((\varphi \wedge \psi))$ (e analogamente para outros operadores binários).

Exemplo 0.2. *Toda fórmula $\varphi \in \mathcal{F}$ tem apenas um número finito de subfórmulas.*

Demonstração. Provamos a afirmação “ φ tem apenas um número finito de subfórmulas” para toda $\varphi \in \mathcal{F}$ por indução estrutural.

Base: Seja $\varphi = p$ atômica. Então o conjunto de suas subfórmulas é $\{ p \}$ e, portanto, finito.

Passo indutivo: Suponhamos que, para φ e ψ , o conjunto de suas respectivas subfórmulas seja finito. Então as subfórmulas de $\neg\varphi$ são dadas por $\{\neg\varphi\} \cup \{\text{subfórmulas de } \varphi\}$, isto é, por uma união finita de conjuntos finitos e, portanto, formam um conjunto finito. Para a fórmula composta $(\varphi \wedge \psi)$, vale analogamente que o conjunto de suas subfórmulas

$$\{ (\varphi \wedge \psi) \} \cup \{\text{subfórmulas de } \varphi\} \cup \{\text{subfórmulas de } \psi\}$$

é uma união finita de conjuntos finitos e, portanto, também é finito.

Logo, por indução estrutural, segue que toda $\varphi \in \mathcal{F}$ possui apenas um número finito de subfórmulas. $\square$

Comparação, observações e casos típicos de aplicação

- **Quando usar qual indução?** Use indução completa quando a afirmação estiver formulada para números naturais (ou diretamente sobre $\mathbb{N}$), por exemplo em combinatória, somas ou sequências recursivas. Use indução estrutural quando as afirmações disserem respeito a objetos definidos recursivamente e com estrutura arbórea, como termos, fórmulas, árvores de prova ou árvores sintáticas abstratas.
- **Forma do passo indutivo.** Na indução completa, o passo indutivo normalmente se refere ao predecessor imediato $n - 1$ ou a todos os $k < n$ (indução forte). Na indução estrutural, mostra-se que a propriedade é preservada pelas regras de construção.
- **Pontos em comum.** Ambos os métodos são aplicações do princípio de que não existem sequências infinitas estritamente decrescentes em uma relação bem fundada. Formalmente, ambos podem ser entendidos como casos especiais da indução bem fundada (*well-founded induction*).
- **Dicas práticas.** Formule as afirmações indutivas de modo que a hipótese de indução contenha exatamente as informações necessárias. Na indução estrutural, muitas vezes é recomendável escolher

afirmações indutivas fortes (por exemplo, afirmações sobre o conjunto de todas as subestruturas), para que as provas nos casos compostos se tornem mais diretas.

0.3 Estrutura e estilo de textos matemáticos

Os textos matemáticos deste livro seguem uma forma clara e reconhecível: definições e notações fixam conceitos e convenções precisos, exemplos ilustram a intuição, resultados (lemas, proposições, teoremas, corolários) formulam as afirmações centrais, e demonstrações justificam essas afirmações de modo rigoroso. Essa estrutura ajuda a leitora ou o leitor a distinguir rapidamente entre enunciado formal, demonstração e material ilustrativo.

Definições e notações

- **Definições.** Definições formalizam novos conceitos. Elas devem ser concisas, completas e escolhidas de tal modo que as propriedades necessárias possam ser verificadas diretamente.
- **Notações.** Notações reúnem formas abreviadas de escrita (por exemplo, $\mathcal{P}(X)$, $X \times Y$, $O(\cdot)$). As notações devem ser uniformes e reunidas em um lugar de destaque (no início do capítulo ou em um quadro de notações).

Exemplos e ilustrações

Exemplos servem à visualização intuitiva e mostram casos típicos ou contrastantes. Eles devem ser breves, demonstrar o uso de novas definições e, sempre que possível, remeter a exercícios ou aplicações posteriores.

Resultados: lema, proposição, teorema, corolário

- **Lema.** Um resultado auxiliar, usado principalmente na demonstração de outro resultado.
- **Proposição.** Um resultado útil, em geral menos central.
- **Teorema.** Um resultado central e importante.
- **Corolário.** Segue imediatamente de um resultado anterior.

Todos os resultados devem conter uma formulação precisa, quando pertinente as hipóteses, e uma demonstração clara.

Demonstrações

Demonstrações fazem parte da argumentação matemática usual. Demonstrações devem ser, tanto quanto possível, bem estruturadas:

1. Uma breve indicação inicial do que será demonstrado.

2. Se necessário: divisão em casos ou em lemas auxiliares.
3. Ao final, um sinal claro de conclusão (geralmente escrito como $\square$).

Observações, exercícios e indicações bibliográficas

- **Observações** contêm frequentemente informações adicionais, intuições ou generalizações. Elas são mais breves e menos formais.
- **Exercícios** ao final do capítulo consolidam a compreensão.
- **Indicações bibliográficas** remetem a fontes complementares e devem ser bibliograficamente precisas.

Breve exemplo-modelo

Definição 0.3. *Uma aplicação $f : X \rightarrow Y$ chama-se injetiva se de $f(x) = f(x')$ segue necessariamente $x = x'$.*

Teorema 0.4. *Seja $f : X \rightarrow Y$ injetiva e seja $g : Y \rightarrow Z$ arbitrária. Então $g \circ f$ é injetiva se, e somente se, g é injetiva sobre a imagem de f .*

Demonstração. A afirmação segue por verificação direta da definição: se $g \circ f$ é injetiva e $g(f(x)) = g(f(x'))$, então vale $f(x) = f(x')$ e, pela injetividade de f , segue $x = x'$. A outra direção é igualmente direta. $\square$

Capítulo 1

Lógica proposicional

1.1 Introdução e visão geral

A lógica proposicional é o sistema formal fundamental por meio do qual enunciados simples e suas conexões lógicas podem ser descritos com precisão. No centro estão as proposições atômicas, cada uma das quais pode assumir exatamente um de dois valores de verdade: verdadeiro ou falso. A partir desses blocos atômicos, formam-se, por meio de conectivos como *e*, *ou* e *não*, fórmulas compostas com as quais também se podem expressar relações lógicas mais complexas.

Apesar de sua sintaxe elementar, a lógica proposicional possui grande alcance. Ela constitui uma base importante para muitas outras áreas da lógica e, ao mesmo tempo, desempenha um papel central na informática, por exemplo na demonstração automática, na

análise de circuitos lógicos, na verificação e em métodos da inteligência artificial. É particularmente útil o fato de a lógica proposicional ser completamente formalizada: tanto a formação das fórmulas quanto seu significado podem ser especificados de modo exato.

Este capítulo introduz inicialmente os fundamentos sintáticos e semânticos. Começamos com assinaturas, fórmulas e interpretações e, em seguida, esclarecemos quando uma fórmula é verdadeira sob uma interpretação e o que significa dizer que um conjunto de fórmulas possui um modelo. Depois consideramos o conceito de consequência lógica, bem como a questão de quando uma fórmula segue de outra ou de um conjunto de fórmulas. Com base nisso, investigamos a equivalência lógica de fórmulas e apresentamos regras importantes de transformação, bem como formas normais, que são especialmente relevantes para o processamento algorítmico.

No decorrer do capítulo, voltamo-nos para procedimentos que não apenas descrevem o raciocínio lógico semanticamente, mas também o executam sintaticamente. Entre eles estão cálculos gerais, assim como o cálculo de resolução, que constitui uma ferramenta central da demonstração automática. Por fim, examinamos mais de perto o lado algorítmico da lógica proposicional: em particular, investigamos o problema da satisfatibilidade, mostramos por que ele é teoricamente difícil e conhecemos, com a codificação de Tseitin e o algoritmo DPLL, dois métodos importantes com os quais a satisfatibilidade pode, na prática, ser verificada muitas vezes de modo

bastante eficaz.

O capítulo está organizado da seguinte maneira: na subseção 1.1, introduzimos a sintaxe e a semântica da lógica proposicional. Tratamos primeiro de assinaturas, depois de fórmulas, interpretações e modelos. Na subseção 1.2, abordamos propriedades de modelos, o conceito de consequência e a independência da assinatura. Na subseção 1.3, tratamos do tema da equivalência; isso inclui tanto a definição de equivalência lógica quanto transformações equivalentes importantes e a representação de proposições em formas normais. Na subseção 1.4, são apresentados diferentes cálculos para a consequência automática, em especial o cálculo de resolução. Por fim, a subseção 1.5 explica a redução a problemas de satisfatibilidade, a codificação de Tseitin e o algoritmo DPLL como métodos centrais para a solução do problema da satisfatibilidade.

Assim, este capítulo conecta a base conceitual da lógica proposicional a seus procedimentos algorítmicos centrais. Com isso, estabelece a base para tudo o que será necessário, ao longo do livro, sobre raciocínio formal, formas normais e procedimentos automáticos de demonstração.

1.2 Sintaxe e semântica

1.2.1 Introdução

Como toda lógica, a lógica proposicional é definida por dois componentes:

- *Sintaxe*: a descrição das expressões bem formadas (fórmulas), isto é, quais sequências de símbolos são consideradas objetos válidos, e
- *Semântica*: a atribuição de significados (valores de verdade) a essas expressões.

No que segue, introduzimos primeiro assinaturas e fórmulas e, em seguida, definimos a semântica modeloteórica.

1.2.2 Assinaturas

Definição 1.1. *Uma assinatura proposicional Σ é um conjunto (finito) de variáveis proposicionais (também chamadas de átomos ou proposições).*

Exemplo 1.2. *Para modelar um sistema de alarme de casa inteligente, pode-se escolher, por exemplo,*

$$\Sigma_{\text{Alarme}} = \{Fumaca, Calor, Movimento, Alarme, Aspersor\},$$

em que os significados pretendidos das variáveis são, por exemplo:

- *Fumaca*: fumaça é detectada
- *Calor*: temperatura elevada é detectada

- *Movimento: movimento na casa é registrado*
- *Alarme: o alarme está disparado*
- *Aspersor: os aspersores estão ativados*

Uma fórmula típica em $L(\Sigma_{\text{Alarme}})$ seria, por exemplo,

$$(Fumaca \vee Calor) \rightarrow Alarme,$$

que afirma: “Se fumaça ou temperatura elevada é detectada, então o alarme é disparado”. ■

1.2.3 Fórmulas

Definição 1.3. *Seja Σ uma assinatura proposicional. O conjunto $L(\Sigma)$ das fórmulas sobre Σ é definido indutivamente por:*

1. *toda variável proposicional $p \in \Sigma$ é uma fórmula (fórmula atômica);*
2. *se φ é uma fórmula, então $\neg\varphi$ é uma fórmula;*
3. *se φ e ψ são fórmulas, então $(\varphi \wedge \psi)$ e $(\varphi \vee \psi)$ são fórmulas.*

Alternativamente (BNF):

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \quad (p \in \Sigma).$$

Notação 1.4. *Usamos as seguintes abreviações e convenções:*

- *Conectivos: $\neg$ (negação), $\wedge$ (conjunção), $\vee$ (disjunção).*

- A implicação e a equivalência são definidas por $\varphi \rightarrow \psi := \neg\varphi \vee \psi$ e $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.
- Convenção de parênteses: $\neg$ tem precedência sobre $\wedge$, que tem precedência sobre $\vee$, que tem precedência sobre $\rightarrow$, que tem precedência sobre $\leftrightarrow$.
- Um literal é ou um átomo p ou sua negação $\neg p$.

Exemplo 1.5. Com Σ_{Alarme} do exemplo acima, as seguintes expressões, por exemplo, são fórmulas em $L(\Sigma_{\text{Alarme}})$:

$$\begin{aligned} & (Fumaca \vee Calor) \rightarrow Alarme, \\ & \neg Fumaca \wedge (Alarme \vee Aspensor), \\ & Movimento \rightarrow Alarme. \end{aligned}$$

O conjunto dos literais é

$$\begin{aligned} \text{Lit}(\Sigma_{\text{Alarme}}) = \{ & Fumaca, \neg Fumaca, Calor, \neg Calor, \\ & Movimento, \neg Movimento, Alarme, \neg Alarme, \\ & Aspensor, \neg Aspensor \}. \end{aligned}$$



1.2.4 Semântica: interpretações, satisfação e modelos

Definição 1.6. Seja Σ uma assinatura. Uma interpretação (ou valoração) ω sobre Σ é uma função $\omega : \Sigma \rightarrow \{\text{verdadeiro}, \text{falso}\}$. Chamamos de $\Omega(\Sigma)$ o conjunto de todas as interpretações.

Definição 1.7. A relação de satisfação $\models_{\text{LP}} \subseteq \Omega(\Sigma) \times L(\Sigma)$ é definida indutivamente:

$$\begin{aligned} \omega \models_{\text{LP}} p & \quad \text{sse} \quad \omega(p) = \text{verdadeiro}, \\ \omega \models_{\text{LP}} \neg \varphi & \quad \text{sse} \quad \text{não } (\omega \models_{\text{LP}} \varphi), \\ \omega \models_{\text{LP}} \varphi \wedge \psi & \quad \text{sse} \quad \omega \models_{\text{LP}} \varphi \text{ e } \omega \models_{\text{LP}} \psi, \\ \omega \models_{\text{LP}} \varphi \vee \psi & \quad \text{sse} \quad \omega \models_{\text{LP}} \varphi \text{ ou } \omega \models_{\text{LP}} \psi. \end{aligned}$$

Se $\omega \models_{\text{LP}} \varphi$, então ω é chamado de modelo de φ .

Definição 1.8. Para uma fórmula φ , definimos

$$\text{Mod}_{\text{LP}}(\varphi) = \{ \omega \in \Omega(\Sigma) \mid \omega \models_{\text{LP}} \varphi \},$$

o conjunto de todos os modelos de φ . Além disso, φ é satisfatível sse $\text{Mod}_{\text{LP}}(\varphi) \neq \emptyset$, e φ é válida (tautologia) sse $\text{Mod}_{\text{LP}}(\varphi) = \Omega(\Sigma)$.

Explicação. Se φ é uma fórmula e ω é um modelo de φ ($\omega \in \text{Mod}_{\text{LP}}(\varphi)$), então ω pode ser entendido intuitivamente como um *mundo possível* ou como uma *situação* na qual a afirmação φ de fato se verifica. A interpretação ω determina, para cada enunciado atômico $p \in \Sigma$, se p é verdadeiro ou falso nesse mundo; desse modo, ω explica *completamente* por que φ é verdadeira: pela atribuição de valores de verdade aos átomos a partir dos quais φ é construída.

A designação *modelo* enfatiza exatamente esse caráter explicativo: um modelo é um testemunho concreto que prova que uma fórmula é satisfatível. Correspondentemente, o conjunto $\Omega(\Sigma)$ é o conjunto de todos os mundos

possíveis (todas as atribuições possíveis), e $\text{Mod}_{\text{LP}}(\varphi)$ consiste naqueles mundos que tornam φ verdadeira.

Dois conceitos breves e úteis:

- *Satisfatibilidade*: A fórmula φ é satisfatível se, e somente se, $\text{Mod}_{\text{LP}}(\varphi) \neq \emptyset$. Um $\omega \in \text{Mod}_{\text{LP}}(\varphi)$ concreto também é chamado de *modelo* ou de *atribuição satisfazente* de φ .
- *Contramodelo*: Para mostrar que uma fórmula φ não é válida (não é tautológica), basta indicar um $\omega \in \Omega(\Sigma)$ tal que $\omega \notin \text{Mod}_{\text{LP}}(\varphi)$; tal ω é chamado de *contramodelo*.

Observação 1.9. *Em aplicações (por exemplo, em model checking), encontrar um modelo corresponde frequentemente à demonstração de que um comportamento desejado do sistema pode ocorrer, enquanto fornecer um contramodelo mostra que uma propriedade desejada é violada sob determinadas atribuições.*

Exemplo 1.10. *Considere Σ_{Alarme} e a interpretação ω com $\omega(\text{Fumaca}) = \text{true}$, $\omega(\text{Calor}) = \text{false}$, $\omega(\text{Movimento}) = \text{true}$, $\omega(\text{Alarme}) = \text{true}$, $\omega(\text{Aspersor}) = \text{false}$. Então vale, por exemplo, $\omega \models_{\text{LP}} \text{Fumaca}$, $\omega \models_{\text{LP}} \neg \text{Calor}$, $\omega \models_{\text{LP}} (\text{Fumaca} \vee \text{Calor}) \rightarrow \text{Alarme}$ (pois $\omega(\text{Fumaca}) = \text{true}$ e $\omega(\text{Alarme}) = \text{true}$), $\omega \models_{\text{LP}} \text{Movimento} \rightarrow \text{Alarme}$ (pois $\omega(\text{Movimento}) = \text{true}$ e $\omega(\text{Alarme}) = \text{true}$) e $\omega \models_{\text{LP}} \neg \text{Aspersor}$. ■*

1.2.5 Tabelas-verdade

Os conectivos podem ser descritos por meio de tabelas-verdade; segue a representação compacta (1 = verdadeiro, 0 = falso):

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\neg\varphi$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Observação 1.11. • Se $|\Sigma| = n$, então $|\Omega(\Sigma)| = 2^n$ interpretações.

- A indução definicional da semântica (satisfação) é equivalente à especificação direta de uma função de verdade $\llbracket \cdot \rrbracket_\omega$.
- A semântica modeloteórica é a base para conceitos como consequência lógica $\Phi \models \psi$ (toda interpretação que satisfaz todas as fórmulas em Φ também satisfaz ψ).

1.2.6 Funções de verdade

Definição 1.12. Seja Σ uma assinatura da lógica proposicional e $\omega \in \Omega(\Sigma)$ uma interpretação. A função de verdade $\llbracket \cdot \rrbracket_\omega : L(\Sigma) \rightarrow \{\text{true}, \text{false}\}$ é definida indutiva-

mente da seguinte forma:

$$\llbracket p \rrbracket_\omega = \omega(p) \quad (p \in \Sigma),$$

$$\llbracket \neg \varphi \rrbracket_\omega = \begin{cases} \text{true}, & \text{se } \llbracket \varphi \rrbracket_\omega = \text{false}, \\ \text{false}, & \text{caso contrário}, \end{cases}$$

$$\llbracket \varphi \wedge \psi \rrbracket_\omega = \begin{cases} \text{true}, & \text{se } \llbracket \varphi \rrbracket_\omega = \text{true} \text{ e } \llbracket \psi \rrbracket_\omega = \text{true}, \\ \text{false}, & \text{caso contrário}, \end{cases}$$

$$\llbracket \varphi \vee \psi \rrbracket_\omega = \begin{cases} \text{false}, & \text{se } \llbracket \varphi \rrbracket_\omega = \text{false} \text{ e } \llbracket \psi \rrbracket_\omega = \text{false}, \\ \text{true}, & \text{caso contrário}. \end{cases}$$

Proposição 1.13. *Seja Σ uma assinatura, $\omega \in \Omega(\Sigma)$ uma interpretação e $\varphi \in L(\Sigma)$. Então vale*

$$\omega \models_{\text{LP}} \varphi \quad \text{sse} \quad \llbracket \varphi \rrbracket_\omega = \text{true}.$$

Demonstração. Seja Σ uma assinatura da lógica proposicional, $\omega \in \Omega(\Sigma)$ uma interpretação e $\varphi \in L(\Sigma)$ uma fórmula. Mostramos a equivalência

$$\omega \models_{\text{LP}} \varphi \quad \text{sse} \quad \llbracket \varphi \rrbracket_\omega = \text{true}$$

por indução estrutural sobre a fórmula φ .

Base da indução. Seja $\varphi = p$, com $p \in \Sigma$, um átomo. Pela definição da relação de satisfação, vale $\omega \models_{\text{LP}} p$ se, e somente se, $\omega(p) = \text{true}$. Pela definição da função de verdade, temos $\llbracket p \rrbracket_\omega = \omega(p)$. Portanto, segue imediatamente que $\omega \models_{\text{LP}} p \iff \llbracket p \rrbracket_\omega = \text{true}$.

Passo indutivo. Seja agora φ uma fórmula, e suponha que a afirmação já valha para todas as subfórmulas próprias de φ (hipótese de indução). Consideramos os diferentes casos de construção.

- $\varphi = \neg\psi$.
 $\Rightarrow$: Suponha que $\omega \models_{\text{LP}} \neg\psi$. Então, pela definição, $\omega \not\models_{\text{LP}} \psi$. Pela hipótese de indução, segue que $\llbracket \psi \rrbracket_{\omega} \neq \text{true}$, ou seja, $\llbracket \psi \rrbracket_{\omega} = \text{false}$. Pela definição da função de verdade, temos então $\llbracket \neg\psi \rrbracket_{\omega} = \text{true}$.
 $\Leftarrow$: Suponha que $\llbracket \neg\psi \rrbracket_{\omega} = \text{true}$. Então $\llbracket \psi \rrbracket_{\omega} = \text{false}$. Pela hipótese de indução, segue que $\omega \not\models_{\text{LP}} \psi$ e, portanto, $\omega \models_{\text{LP}} \neg\psi$.
- $\varphi = \psi \wedge \xi$.
 $\Rightarrow$: Suponha que $\omega \models_{\text{LP}} \psi \wedge \xi$. Então vale $\omega \models_{\text{LP}} \psi$ e $\omega \models_{\text{LP}} \xi$. Pela hipótese de indução, temos, portanto, $\llbracket \psi \rrbracket_{\omega} = \text{true}$ e $\llbracket \xi \rrbracket_{\omega} = \text{true}$. Logo, $\llbracket \psi \wedge \xi \rrbracket_{\omega} = \text{true}$.
 $\Leftarrow$: Suponha que $\llbracket \psi \wedge \xi \rrbracket_{\omega} = \text{true}$. Então $\llbracket \psi \rrbracket_{\omega} = \text{true}$ e $\llbracket \xi \rrbracket_{\omega} = \text{true}$. Pela hipótese de indução, segue que $\omega \models_{\text{LP}} \psi$ e $\omega \models_{\text{LP}} \xi$, logo $\omega \models_{\text{LP}} \psi \wedge \xi$.
- $\varphi = \psi \vee \xi$.
 $\Rightarrow$: Suponha que $\omega \models_{\text{LP}} \psi \vee \xi$. Então vale $\omega \models_{\text{LP}} \psi$ ou $\omega \models_{\text{LP}} \xi$. Sem perda de generalidade, seja $\omega \models_{\text{LP}} \psi$. Pela hipótese de indução, temos $\llbracket \psi \rrbracket_{\omega} = \text{true}$, de modo que $\llbracket \psi \vee \xi \rrbracket_{\omega} = \text{true}$. (O outro caso é análogo.)
 $\Leftarrow$: Suponha que $\llbracket \psi \vee \xi \rrbracket_{\omega} = \text{true}$. Então $\llbracket \psi \rrbracket_{\omega} =$

true ou $\llbracket \xi \rrbracket_\omega = \text{true}$. Pela hipótese de indução, segue que $\omega \models_{\text{LP}} \psi$ ou $\omega \models_{\text{LP}} \xi$, e, portanto, $\omega \models_{\text{LP}} \psi \vee \xi$.

Como todos os casos de construção foram cobertos, a afirmação segue para toda fórmula φ . $\square$

Os conectivos também podem ser apresentados de forma compacta por meio de tabelas de verdade. Primeiro, a variante com os rótulos true/false:

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\neg\varphi$
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

E aqui a representação compacta em 0/1 (0 = false, 1 = true):

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\neg\varphi$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Observação. A definição acima da função de verdade é equivalente à definição indutiva da relação de satisfação $\models_{\text{LP}}$. Em demonstrações, uma ou outra formulação pode ser mais conveniente conforme o contexto: a relação de

satisfação enfatiza a perspectiva dos modelos ($\omega \models \varphi$), enquanto a função de verdade indica explicitamente o valor da fórmula em uma interpretação dada ($\llbracket \varphi \rrbracket_\omega \in \{\text{true}, \text{false}\}$).

Perspectiva

Na próxima seção, trataremos das formas normais sintáticas, das regras concretas de transformação (por exemplo, FNC, FND) e dos primeiros procedimentos de prova (resolução, dedução natural), que são relevantes tanto do ponto de vista teórico quanto do ponto de vista prático (por exemplo, para SAT-solvers).

1.3 Exercícios com soluções

Exercício 1. Seja a assinatura proposicional $\Sigma = \{P, Q, R, S, T, U\}$. Sejam dados os seguintes objetos:

$$\varphi_1 = S \wedge \neg(P \leftrightarrow \neg R),$$

$$\omega_2 : P \mapsto \text{true}, Q \mapsto \text{false}, R \mapsto \text{true},$$

$$S \mapsto \text{false}, T \mapsto \text{true}, U \mapsto \text{false}.$$

e

$$\varphi_3 = \neg T \wedge \neg(\neg T \vee R).$$

Decida, para as seguintes afirmações, se elas são **verdadeiras** ou **falsas** (justifique brevemente em cada caso):

1. $\neg P \vee R \in L(\Sigma)$.

2. A fórmula indicada

$$\overline{\varphi_1} = \neg S \vee (\neg P \vee \neg R) \wedge (R \vee P) \vee \neg Q$$

é o complemento (a negação) de φ_1 .

3. $\llbracket \neg T \vee (\neg R \leftrightarrow P) \rrbracket_{\omega_2} = \text{false}$.
 4. $|\text{Mod}_{\text{LP}}(\varphi_3)| = 0$.

Soluções e explicações.

1. **Verdadeiro.** $\neg P \vee R$ é uma expressão bem formada, construída a partir de átomos de Σ com os conectivos $\neg$ e $\vee$, portanto é uma fórmula em $L(\Sigma)$.
 2. **Falso.** A fórmula afirmada não é a negação de φ_1 . Em primeiro lugar, ela contém o átomo Q , que não ocorre em φ_1 ; em segundo lugar, a estrutura sintática de uma negação de $\varphi_1 = S \wedge \neg(P \leftrightarrow \neg R)$ é, pelas leis de De Morgan, equivalente a

$$\neg S \vee \neg \neg(P \leftrightarrow \neg R) = \neg S \vee (P \leftrightarrow \neg R),$$

e não à fórmula complicada indicada, com $\neg Q$ e o termo adicional $(\neg P \vee \neg R) \wedge (R \vee P)$.

3. **Verdadeiro.** Calculamos a fórmula $\neg T \vee (\neg R \leftrightarrow P)$ sob ω_2 . Sob ω_2 , vale $T = \text{true}$, logo $\neg T = \text{false}$. Além disso, $R = \text{true}$, portanto $\neg R = \text{false}$. Como $P = \text{true}$, a equivalência proposicional $\neg R \leftrightarrow P$ entre false e true tem valor false . Assim, $\neg T \vee (\neg R \leftrightarrow P)$ é igual a $\text{false} \vee \text{false} = \text{false}$. Portanto, a afirmação está correta.

4. **Verdadeiro.** Simplificamos $\varphi_3 = \neg T \wedge \neg(\neg T \vee R)$. A subfórmula $\neg(\neg T \vee R)$ é equivalente a $\neg\neg T \wedge \neg R$, isto é, a $T \wedge \neg R$. Portanto, $\varphi_3 \equiv \neg T \wedge (T \wedge \neg R)$. Essa fórmula é evidentemente insatisfatível, pois exigiria simultaneamente $\neg T$ e T . Logo, φ_3 não tem *nenhum* modelo e, assim, $|\text{Mod}_{\text{LP}}(\varphi_3)| = 0$.

Comentários:

- (1) A fórmula pertence à linguagem sobre Σ — correto.
- (2) A fórmula de negação indicada está incorreta — ela contém um átomo estranho e não possui a estrutura correta (não é equivalente à negação de φ_1).
- (3) Verdadeiro — a avaliação resulta em false.
- (4) Verdadeiro — φ_3 é contraditória e, portanto, insatisfatível.

Exercício ampliado. Varie a assinatura Σ (por exemplo, usando apenas três átomos) e determine, para todas as $2^{|\Sigma|}$ interpretações, quais fórmulas da Tarefa 1 são verdadeiras. Compare a quantidade de modelos das respectivas fórmulas.

Solução-modelo (exemplo com $\Sigma' = \{P, Q, R\}$)

Escolhemos $\Sigma' = \{P, Q, R\}$ e definimos as fórmulas análogas às da tarefa (adaptação do exercício original à assinatura menor):

$$\varphi_1 := Q \wedge \neg(P \leftrightarrow \neg R),$$

$$\alpha := \neg P \vee R,$$

$$\beta := \neg Q \vee ((\neg P \vee \neg R) \wedge (R \vee P)) \quad (\text{candidata a } \neg\varphi_1),$$

$$\gamma := \neg R \vee (\neg P \leftrightarrow Q),$$

$$\delta := \neg P \wedge \neg(\neg P \vee R).$$

A fórmula $(\neg P \vee \neg R) \wedge (R \vee P)$ é equivalente a $(P \leftrightarrow \neg R)$, de modo que β representa exatamente $\neg Q \vee (P \leftrightarrow \neg R)$ — isto é, formalmente, a negação de $\varphi_1 = Q \wedge \neg(P \leftrightarrow \neg R)$.

A tabela-verdade completa (T = true, F = false):

P	Q	R	φ_1	α	β	γ	δ
F	F	F	F	T	T	T	F
F	F	T	F	T	T	F	F
F	T	F	T	T	F	T	F
F	T	T	F	T	T	T	F
T	F	F	F	F	T	T	F
T	F	T	F	T	T	T	F
T	T	F	F	F	T	T	F
T	T	T	T	T	F	F	F

A partir da tabela, obtemos o número de modelos (número de interpretações para as quais a fórmula é verdadeira):

- $|\text{Mod}(\varphi_1)| = 2$ (verdadeira para as atribuições $(P, Q, R) = (F, T, F)$ e (T, T, T))

- $|\text{Mod}(\alpha)| = 6$
- $|\text{Mod}(\beta)| = 6$
- $|\text{Mod}(\gamma)| = 6$
- $|\text{Mod}(\delta)| = 0$ (insatisfatível)

Interpretação e conclusões

- $\alpha = \neg P \vee R$ não é tautológica (não é verdadeira em todas as interpretações), mas é satisfeita em 6 de 8 casos — portanto, é frequente, mas não universalmente válida.
- β , nesta formulação concreta, é de fato logicamente equivalente a $\neg\varphi_1$. Isso pode ser visto algebricamente:

$$\begin{aligned}\neg\varphi_1 &= \neg(Q \wedge \neg(P \leftrightarrow \neg R)) \\ &= \neg Q \vee \neg\neg(P \leftrightarrow \neg R) \\ &= \neg Q \vee (P \leftrightarrow \neg R)\end{aligned}$$

e $P \leftrightarrow \neg R$ é equivalente a $(\neg P \vee \neg R) \wedge (R \vee P)$. Portanto, β e $\neg\varphi_1$ têm os mesmos modelos (6 interpretações).

- γ também é verdadeira em 6 interpretações; seu conjunto de modelos difere daquele de α e β em sua composição exata, mas as cardinalidades são iguais (6).

- δ é contraditória: $\delta = \neg P \wedge (P \wedge \neg R)$ (após a transformação) exige simultaneamente P e $\neg P$ — portanto, não possui modelos.

Observação sobre o procedimento (generalização)

Para assinaturas arbitrárias Σ com n átomos, pode-se listar de modo análogo todas as 2^n interpretações (ou gerá-las programaticamente) e avaliar cada fórmula automaticamente. O número de modelos de uma fórmula é exatamente o número de linhas da tabela-verdade ou da tabela de avaliação nas quais a fórmula assume o valor true. Para exercícios manuais com $n = 3$ ou $n = 4$, isso é praticamente viável; para valores maiores de n , recomenda-se usar um pequeno script (ou um SAT-solver) para contar os modelos.

1.4 Inferência

Na prática, utilizamos fórmulas lógicas não apenas para descrever estados isolados, mas sobretudo para obter novos conhecimentos a partir de conhecimentos já dados. Por *inferir*, entendemos formalmente a seguinte questão: quais afirmações devem valer em todos os mundos nos quais nossas hipóteses são verdadeiras?

Este capítulo define a noção semântica de consequência, apresenta regras simples de cálculo para conjuntos de modelos e relaciona entre si os conceitos de “modelo”, “teoria” e “fecho dedutivo”.

1.4.1 Modelos como descrição de situações

Recordemos: para uma assinatura Σ , $\Omega(\Sigma)$ é o conjunto de todas as valorações possíveis das variáveis atômicas. A uma fórmula φ corresponde o conjunto de modelos

$$\mathcal{M}(\varphi) = \{\omega \in \Omega(\Sigma) \mid \omega \models_{\text{LP}} \varphi\},$$

isto é, aquelas valorações que tornam φ verdadeira.

Intuição. Pode-se pensar em $\omega \in \mathcal{M}(\varphi)$ como uma “situação” concreta na qual a descrição φ se aplica. Assim, um modelo explica completamente por que φ é verdadeira: pelos valores de verdade de seus componentes atômicos.

1.4.2 Operações sobre conjuntos de modelos

Os conectivos clássicos correspondem a operações simples sobre conjuntos de modelos. Essas relações são úteis porque permitem transferir argumentos semânticos para o nível das operações com conjuntos.

Proposição 1.14. *Para todas as fórmulas $\varphi, \psi \in L(\Sigma)$, vale*

$$\mathcal{M}(\varphi \wedge \psi) = \mathcal{M}(\varphi) \cap \mathcal{M}(\psi), \quad \mathcal{M}(\varphi \vee \psi) = \mathcal{M}(\varphi) \cup \mathcal{M}(\psi),$$

$$\mathcal{M}(\neg \varphi) = \Omega(\Sigma) \setminus \mathcal{M}(\varphi).$$

Demonstração. Seja Σ uma assinatura da lógica proposicional e sejam $\varphi, \psi \in L(\Sigma)$ fórmulas. Mostramos as três

identidades separadamente, em cada caso provando as duas inclusões.

(1) $\mathcal{M}(\varphi \wedge \psi) = \mathcal{M}(\varphi) \cap \mathcal{M}(\psi)$.

“ $\subseteq$ ”: Seja $\omega \in \mathcal{M}(\varphi \wedge \psi)$. Então vale $\omega \models_{\text{LP}} \varphi \wedge \psi$. Pela definição semântica da conjunção, isso significa que valem tanto $\omega \models_{\text{LP}} \varphi$ quanto $\omega \models_{\text{LP}} \psi$. Logo, ω pertence a $\mathcal{M}(\varphi)$ e a $\mathcal{M}(\psi)$; portanto, $\omega \in \mathcal{M}(\varphi) \cap \mathcal{M}(\psi)$.

“ $\supseteq$ ”: Seja $\omega \in \mathcal{M}(\varphi) \cap \mathcal{M}(\psi)$. Então valem $\omega \models_{\text{LP}} \varphi$ e $\omega \models_{\text{LP}} \psi$. Pela definição semântica da conjunção, segue daí que $\omega \models_{\text{LP}} \varphi \wedge \psi$, isto é, $\omega \in \mathcal{M}(\varphi \wedge \psi)$.

Das duas direções segue a igualdade desejada.

(2) $\mathcal{M}(\varphi \vee \psi) = \mathcal{M}(\varphi) \cup \mathcal{M}(\psi)$.

“ $\subseteq$ ”: Seja $\omega \in \mathcal{M}(\varphi \vee \psi)$. Então $\omega \models_{\text{LP}} \varphi \vee \psi$, isto é, $\omega \models_{\text{LP}} \varphi$ ou $\omega \models_{\text{LP}} \psi$. No primeiro caso, $\omega \in \mathcal{M}(\varphi)$; no segundo caso, $\omega \in \mathcal{M}(\psi)$. Em ambos os casos vale $\omega \in \mathcal{M}(\varphi) \cup \mathcal{M}(\psi)$.

“ $\supseteq$ ”: Seja $\omega \in \mathcal{M}(\varphi) \cup \mathcal{M}(\psi)$. Então vale $\omega \models_{\text{LP}} \varphi$ ou $\omega \models_{\text{LP}} \psi$. Em ambos os casos segue imediatamente $\omega \models_{\text{LP}} \varphi \vee \psi$ e, portanto, $\omega \in \mathcal{M}(\varphi \vee \psi)$.

Com isso, esta igualdade também está demonstrada.

(3) $\mathcal{M}(\neg\varphi) = \Omega(\Sigma) \setminus \mathcal{M}(\varphi)$.

Mostramos as duas afirmações $\Omega(\Sigma) = \mathcal{M}(\varphi) \cup \mathcal{M}(\neg\varphi)$ e $\mathcal{M}(\varphi) \cap \mathcal{M}(\neg\varphi) = \emptyset$, das quais segue a afirmação.

Primeiro, seja $\omega \in \Omega(\Sigma)$ arbitrário. Ou $\omega \models_{\text{LP}} \varphi$, ou não. No primeiro caso, $\omega \in \mathcal{M}(\varphi)$. No segundo caso, vale $\omega \models_{\text{LP}} \neg\varphi$ e, portanto, $\omega \in \mathcal{M}(\neg\varphi)$. Assim, $\Omega(\Sigma) \subseteq \mathcal{M}(\varphi) \cup \mathcal{M}(\neg\varphi)$; a inclusão inversa é trivial, logo há igualdade.