Lógica e aplicações: Matemática, Ciência da Computação e Filosofia (Versão Preliminar - Capítulos 1 a 5)

W.A. Carnielli[1], M.E. Coniglio[1] e R. Bianconi[2]
[1]Departamento de Filosofia
Universidade Estadual de Campinas
C.P. 6133, CEP 13081-970
Campinas, SP, Brasil
E-mail: {carniell,coniglio}@cle.unicamp.br
[2]Instituto de Matemática e Estatística
Universidade de São Paulo
C.P. 66281, CEP 05315-970
São Paulo, SP, Brasil
E-mail: bianconi@ime.usp.br

(Comentários e sugestões são muito bem-vindos)

6 de março de 2006

Sumário

Capítulo 1

Histórico e Paradoxos

1.1 Os Paradoxos Lógicos e o Infinito

O infinito tem sido historicamente a maior fonte de problemas nas ciências formais (matemática, lógica e mais modernamente nas ciências da computação), sendo os dilemas colocados pelo infinito conhecidos desde a antiguidade. Esta questão tem preocupado os filósofos e os matemáticos a tal ponto que o grande matemático alemão David Hilbert em seu conhecido discurso *Über das Unendliche* ("Sobre o Infinito") proferido na cidade de Münster em 1925, chegou a afirmar que "... é portanto o problema do infinito, no sentido acima indicado, que temos que resolver de uma vez por todas".

Hilbert sabia que a presença do infinito ameaça a consistência dos sistemas matemáticos (embora não seja a única causa possível de problemas de fundamentos), e dedicou sua vida a tentar provar que o uso do infinito poderia ser eliminado de uma vez por todas da matemática, dentro do chamado "Programa de Hilbert". Como se sabe, Hilbert não teve sucesso, tendo sido suas pretensões derrotadas pelos Teoremas de Gödel, demonstrados por Kurt Gödel na década de 30.

O infinito é incompreensível à consciência humana, primeiro porque não existe como entidade física (não há no universo nenhum exemplo de alguma classe infinita, e de fato parece ser impossível existir, pelas leis da física e da moderna cosmologia). O infinito só existe na imaginação dos cientistas, que precisam dele basicamente para elaborar teorias com generalidade suficiente para que possam ser interessantes. Por exemplo, se queremos uma teoria simples que possa se referir à aritmética, não podemos supor que exista um último número natural N, pois as operações elementares com valores menores que N claramente ultrapassam N (o resultado do produto de números menores que N pode ser maior que N). Dessa forma, somos obrigados a trabalhar com a hipótese de que a seqüência dos números naturais é *ilimitada*, ou seja, infinita de algum modo.

Este modo de tratar as quantidades ilimitadas corresponde ao chamado *infinito potencial*, isto é, teoricamente não limitado. A esse conceito se contrapõe a idéia do *infinito atual* ou *infinito completado*, quando tratamos uma classe infinita como um todo. Por exemplo, se queremos estudar as propriedades do conjunto $\mathbb{N}$ dos números naturais (o qual, em termos de ordem, também nos referimos como ω) estamos tratando com o infinito atual, ou em outras palavras, assumindo-o como completado. Muitos paradoxos antigos (como os paradoxos de Zenão de Eléia) exploram esta distinção: se assumimos que o infinito atual é possível, o paradoxo se resolve (como discutiremos a seguir). Mas assumir que o infinito atual existe tem seu preço, e cria outros paradoxos, como veremos mais adiante.

Podemos ver o mecanismo de prova por indução que será bastante usado neste livro (ver Capítulo 2) como um mecanismo que permite passar, dentro da aritmética, do infinito potencial ao infinito atual. Dessa forma, na matemática usual, não precisamos nos preocupar com a distinção entre infinito atual e potencial, embora essa distinção continue a ser um problema filosófico interessante.

Quando assumimos o infinito atual, como mostrou o matemático russo Georg Cantor, criador da moderna teoria dos conjuntos, somos obrigados a admitir que existe não um, mas infinitos tipos de infinito. Por exemplo, Cantor mostrou que a quantidade infinita de números naturais, embora seja a mesma quantidade dos números racionais, é distinta da quantidade infinita de números reais.

A seguir, com finalidade de interessar o leitor sobre o que podemos chamar a Grande Questão do Infinito, vamos mostrar alguns tipos de propriedades e de problemas que ilustram de maneira simples o caráter paradoxal do infinito, antes de passarmos aos outros paradoxos que envolvem mecanismos mais sofisticados (como a auto-referência, que como veremos, encerra de algum modo uma idéia de regresso infinito).

1.2 Algumas Propriedades Paradoxais do Infinito

1.2.1 O Paradoxo de Galileu

Consta do folclore que Galileu Galilei haveria ficado muito intrigado com a seguinte questão: se o conjunto dos números pares está contido propriamente no conjunto dos números naturais, deve haver menos pares que naturais (por exemplo, os ímpares não são pares, e eles são infinitos).

Porém se fizermos a seguinte identificação:

$$\begin{aligned} &2 \mapsto 1 \\ &4 \mapsto 2 \\ &6 \mapsto 3 \\ &\ \vdots \\ &2n \mapsto n \\ &\ \vdots \end{aligned}$$

podemos fazer o conjunto dos pares ocupar todo o conjunto dos naturais.

Como é possível que a parte não seja menor que o todo?

> *Justamente, falhar a propriedade de que o todo seja maior que as partes é uma característica das coleções infinitas, o que explica em parte nossa dificuldade em compreendê-las.*

1.2.2 O Passeio de Cantor e os tipos distintos de infinito

Durante muito tempo (até pelo menos o século XIX) pensou-se que os números racionais não poderiam ser enumeráveis (ou contáveis) como os naturais, uma vez que entre cada dois racionais $\frac{a}{b}$ e $\frac{c}{d}$ existe sempre outro, que é sua média aritmética:

$$\frac{a.d + b.c}{2.b.d}$$

e portanto entre dois racionais há infinitos outros.

Georg Cantor mostrou por um artifício muito simples que, embora realmente entre cada dois racionais haja infinitos outros, basta contarmos os racionais de uma maneira diferente da usual (por "usual" entendemos a ordem dos números reais, vistos como pontos de uma reta) para que possamos nos convencer que há precisamente tantos racionais quanto números naturais. Para isso, estabeleceremos uma enumeração dos números racionais positivos (maiores que zero), e a partir daí, é possível (por um argumento análogo àquele da prova da similaridade entre os pares e os naturais) provar a enumerabilidade dos racionais:

$$
\begin{tikzcd}
1/1 \arrow[r] & 1/2 \arrow[dl] & 1/3 \arrow[r] & 1/4 \arrow[dl] & 1/5 & \cdots \\
2/1 \arrow[d] & 2/2 \arrow[ur] & 2/3 \arrow[dl] & 2/4 \arrow[ur] & 2/5 & \cdots \\
3/1 \arrow[ur] & 3/2 \arrow[dl] & 3/3 \arrow[ur] & 3/4 & 3/5 & \cdots \\
4/1 \arrow[d] & 4/2 \arrow[ur] & 4/3 & 4/4 & 4/5 & \cdots \\
5/1 \arrow[ur] & 5/2 & 5/3 & 5/4 & 5/5 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{tikzcd}
$$

Nessa tabela, mesmo que algumas frações equivalentes apareçam várias vezes (como, por exemplo, $1/1, 2/2, 3/3, \ldots$ etc.), todos os racionais positivos certamente aparecem, e cada um recebe um número natural, como mostra o caminho traçado no diagrama acima (chamado de *passeio de Cantor*).

Por outro lado, os números reais não podem, de fato, ser enumerados: suponhamos, para poder chegar a um absurdo, que os reais sejam enumeráveis. Se assim o fosse, os reais no intervalo fechado $[0, 1]$ também seriam. E dentro dessa suposição imaginemos que temos uma lista completa deles usando a expansão decimal, de modo que se o número não é uma dízima como $0,345$ escrevemo-lo com infinitos zeros à direita $0,345000\ldots$; mais ainda, o número 1 é representado como $0,999\ldots$:

Ordem	*Real*							
1	0,	a_{11}	a_{12}	a_{13}	a_{14}	$\dots$	a_{1n}	$\dots$
2	0,	a_{21}	a_{22}	a_{23}	a_{24}	$\dots$	a_{2n}	$\dots$
3	0,	a_{31}	a_{32}	a_{33}	a_{34}	$\dots$	a_{3n}	$\dots$
4	0,	a_{41}	a_{42}	a_{43}	a_{44}	$\dots$	a_{4n}	$\dots$
$\vdots$					$\vdots$			
m	0,	a_{m1}	a_{m2}	a_{m3}	a_{m4}	$\dots$	a_{mn}	$\dots$
$\vdots$					$\vdots$			

onde cada a_{ij} representa um dígito entre 0 e 9.

Vamos construir um outro número real d em $[0,1]$ que não pertence a esta lista: para cada dígito a_{ij} considere o dígito $\bar{a}_{ij} = a_{ij} + 1$ (definindo, no caso em que $a_{ij} = 9$, o dígito $\bar{a}_{ij} = 0$). Definimos então d como:

$$d = 0,\, \bar{a}_{11}\, \bar{a}_{22}\, \bar{a}_{33} \,\dots\, \bar{a}_{nn} \,\dots$$

Observando que $\bar{a}_{ij} \neq a_{ij}$, é claro que d não está na lista, pois difere de um dígito de cada um dos outros da lista. Com efeito, d difere do primeiro número da lista (ao menos) no primeiro dígito, do segundo número da lista (ao menos) no segundo dígito e, em geral, do n-ésimo número da lista (ao menos) no n-ésimo dígito. Portanto d, um número real entre 0 e 1, difere de todos os números da lista acima, o que é um absurdo, pois havíamos suposto que lista era completa. Assim se prova que não podemos enumerar os reais, que constituem então um conjunto infinito maior (isto é, com mais elementos) que os naturais e racionais.

Chamamos o infinito dos reais de "infinito não enumerável".

Cantor provou que estes são apenas os primeiros de uma quantidade infinita de infinitos, mostrando basicamente que o conjunto das partes de um conjunto infinito de uma dada ordem produz um conjunto infinito de ordem superior.

Mencionamos finalmente que o método usado na demonstração acima da não-enumerabilidade do intervalo real $[0,1]$ é chamado de *método diagonal de Cantor* (observe que o número d é construído modificando a diagonal da matriz infinita definida acima), e é utilizado frequentemente na área da Computabilidade, com as modificações necessárias em cada caso.

1.2.3 O Hotel de Hilbert

Outra brincadeira folclórica é o chamado "Hotel de Hilbert": existe um certo hotel, com um número infinito e enumerável de quartos

$$Q_1, Q_2, Q_3, \dots Q_n, \dots$$

Por sorte do proprietário, os quartos estão todos lotados. No meio da noite chega mais um hóspede sem reserva. O gerente simplesmente pede a cada hóspede que se mude para o quarto da direita, liberando o quarto Q_1 para o viajante inesperado.

Na outra noite chegam dois novos hóspedes, e o gerente pede agora a cada hóspede que se mude dois quartos à direita, liberando os quartos Q_1 e Q_2, e assim por diante. Uma noite porém chega um ônibus de excursão (bastante grande) trazendo infinitos novos hóspedes sem reserva.

O gerente agora pede a cada hóspede que se mude para o quarto cujo número seja o dobro do seu (de Q_n para Q_{2n}), liberando espaço para todos.

Assim ele continua recebendo quantos hóspedes novos quiser, até que uma noite (hóspedes inesperados sempre chegam à noite) estaciona um ônibus da *Cia. Real de Turismo*, e ele se apavora. Por quê?

1.2.4 O Lema de König

O Lema de König é uma forma (válida somente para conjuntos enumeráveis) do chamado Axioma da Escolha, que propõe que todo conjunto pode ser bem-ordenado (isto é, ordenado de forma que quaisquer de seus subconjuntos tenha um primeiro elemento com relação a esta ordem).

O Lema de König afirma que[1]

> *Toda árvore infinita, que seja finitamente gerada (isto é, tal que cada ramo tenha um número finito de descendentes) possui pelo menos um ramo infinito.*

Damos a seguir duas aplicações interessantes do Lema de König :

1. O Problema da Descendência

[1] Os conceitos formais de *árvore*, *ramo*, *descendente*, *nós* e *nós sucessores* serão definidos no Capítulo 4 deste livro.

Se a vida na Terra não se acabar, existe uma pessoa que vai ter infinitos descendentes.

Sugestão: pense numa árvore e use o Lema de König.

2. O Problema da Caixa de Bolas

   Uma caixa contém inicialmente uma bola marcada com um número arbitrário. Imagine que podemos sempre trocar uma bola por uma quantidade qualquer (finita) de bolas, mas marcadas com um número menor. Por exemplo, podemos trocar uma bola marcada com 214 por 1.000.000 de bolas marcadas com 213, e assim por diante. No caso porém de retirarmos uma bola marcada por zero, não colocamos nenhuma outra. Será possível por esse processo esvaziar a caixa?

   Sugestão: pense numa árvore cujos nós sejam as bolas, e cujos nós sucessores sejam as que foram colocadas em seu lugar. Use o Lema de König.

1.3 Os Paradoxos Lógicos

1.3.1 O significado dos paradoxos

Vários exemplos na literatura e na pintura, como os quadros do pintor belga René Magritte e os desenhos do holandês M. C. Escher, fazem uso da noção de auto-referência e de seu caráter paradoxal como elemento de estilo. Numa passagem do "Ulisses" de James Joyce, por exemplo, uma das personagens centrais, Molly Bloom, questiona o próprio autor.

Um dos mais simples, e provavelmente o mais antigo, dos paradoxos lógicos é o "Paradoxo do Mentiroso", formulado por um pensador cretense do século VI A.C., Epimênides, que dizia: "Todos os cretenses são mentirosos".

Esta sentença, só superficialmente problemática, é frequentemente confundida com o paradoxo de Eubulides de Mileto, que afirma "Eu estou mentindo", esta sim, uma afirmação paradoxal e que está na raiz de um dos resultados da lógica formal mais importantes do século XX. A versão de Epimênides figura na Bíblia, tornando a lógica a única disciplina com referência bíblica: "Os cretenses são sempre mentirosos, feras selvagens, glutões preguiçosos", adverte a epístola de São Paulo a Tito (1:12-13), chamando a atenção para o fato de que o próprio cretense Epimênides o afirma.

O paradoxo do mentiroso na versão de Eubulides (na forma "Eu estou mentindo" ou "Esta sentença é falsa"), longe de ser uma simples banalidade do pensamento, está ligado, como veremos, a um dos teoremas mais profundos do pensamento lógico e matemático, o Teorema de Gödel, formulado em 1936.

Pode parecer que a auto-referência é a causa destes paradoxos; contudo, a auto-referência, por si mesma, não é nem sempre responsável pelo caráter paradoxal das asserções, nem mesmo suficiente para causar paradoxos: por exemplo, se um cretense afirma "Os cretenses nunca são mentirosos", ou se Eubulides afirma "Não estou mentindo" estas afirmações auto-referentes são apenas pretensiosas.

Por outro lado, mesmo que abolíssemos a auto-referência não eliminaríamos os paradoxos: por exemplo, um paradoxo conhecido desde a época medieval imagina o seguinte diálogo entre Sócrates e Platão:

Sócrates: "O que Platão vai dizer é falso"

Platão: "Sócrates acaba de dizer uma verdade".

Nenhuma das sentenças pode ser verdadeira, e nem falsa; nesse caso, a causa do paradoxo é a referência cruzada ou circular, e não a auto-referência. Mas nem mesmo a circularidade da referência é sempre responsável pelos paradoxos: uma simples mudança no diálogo entre Sócrates e Platão (basta trocar "falso" por "verdadeiro" e vice-versa) elimina o paradoxo, embora a circularidade continue presente.

Na realidade, um dos problemas lógicos mais difíceis é determinar quais são as condições que geram paradoxos, além das tentativas de eliminar, solucionar ou controlar os já existentes. Este problema é, em muitos casos, insolúvel, e tal fato tem obviamente um enorme significado para o pensamento científico em geral, e para a lógica em particular.

Apresentamos aqui alguns dos mais conhecidos paradoxos, antinomias e círculos viciosos. A análise dos paradoxos serve como motivação ao estudo da lógica matemática, que poderia ser pensada como a formalização do pensamento racional livre de paradoxos, pelo menos dos paradoxos que a destroem.

Se aceitamos as leis básicas da lógica tradicional (isto é, leis que regem os operadores "ou", "e", "se... então", "não", "para todo", "existe"), que são o objeto de estudo deste livro, há basicamente duas maneiras de "resolver" um paradoxo:

1. a primeira (seguindo uma tradição iniciada pelo lógico inglês Bertrand Russell) que propõe que certos enunciados paradoxais deixem de ser considerados como enunciados propriamente ditos;

2. a segunda (a partir de idéias devidas ao lógico polonês Alfred Tarski) propõe que sejam considerados como regulares os enunciados onde não ocorre o predicado de "ser verdade" (ou assemelhados, como "ser falso"); os enunciados linguísticos que não são regulares fazem parte da metalinguagem.

Costuma-se ainda fazer distinção, na literatura, entre paradoxos e antinomias: estas seriam as contradições lógicas, como o Paradoxo de Russell e do Mentiroso, enquanto os paradoxos propriamente ditos seriam os enunciados que não envolvem contradição, mas desafiam nossas intuições ou crenças.

As situações paradoxais apresentadas a seguir são formuladas na linguagem natural (isto é, em português corrente); como exercício, você deve tentar analisá-las, informalmente, e decidir se se trata de antinomia, se existe solução, ou simplesmente de uma situação paradoxal que escapa à intuição.

Os problemas aqui encontrados servirão de motivação para que seja introduzida uma *linguagem formal*, muito mais simples que a linguagem natural, mas também muito mais exata, e que com base nesta linguagem sejam formuladas cuidadosamente as regras e leis que regem a lógica. Dessa forma, podemos então considerar a lógica não como uma teoria que resolve todos os paradoxos, mas como uma disciplina que aprende com eles e que tenta erigir um domínio em que se minimizem seus efeitos.

1.3.2 Paradoxos e antinomias mais conhecidos

1. Numa folha de papel em branco escreva: "A sentença do outro lado é verdadeira". No outro lado escreva: "A sentença do outro lado é falsa". As sentenças são verdadeiras ou falsas?

2. (Paradoxo de Bertrand Russell, numa carta a G. Frege, em 1902) Considere o conjunto de todos os conjuntos que não são membros de si mesmos. Este conjunto é membro de si próprio?

3. (Paradoxo do Barbeiro) Um barbeiro foi condenado a barbear todos e somente aqueles homens que não se barbeiam a si próprios. Quem barbeia o barbeiro?

4. (Paradoxo de Kurt Grelling) Podemos dividir os adjetivos em duas classes: autodescritivos e não-autodescritivos. Por exemplo, são autodescritivos os adjetivos "polissílabo", "escrito", e não-autodescritivos os adjetivos "monossílabo", "verbal", etc. O adjetivo "não-autodescritivo" é autodescritivo ou não-autodescritivo?

5. Qual é “o menor número inteiro que não se pode expressar com menos de quinze palavras” ? (conte quantas palavras expressam este número).

6. Qual é o menor número inteiro que não se menciona de nenhuma maneira nestas notas? Existe tal número?

7. Se não existe, estas notas mencionam todos os números inteiros?

8. Analise a seguinte prova da existência de Deus: escreva “Esta sentença é falsa ou Deus existe”. Se a sentença toda for falsa, as duas partes separadas por “ou” são falsas, portanto a parte “Esta sentença é falsa” é falsa; sendo falso que “Esta sentença é falsa” obriga a que “Esta sentença é falsa” seja uma sentença verdadeira, tornando verdadeira a sentença toda, contradição.

   Portanto a sentença toda é verdadeira, logo uma de suas partes é verdadeira. É claro que a primeira parte não pode ser verdadeira (pois isto contraria o que ela está afirmando, isto é, a sua falsidade), logo a segunda parte deve ser verdadeira, isto é, Deus existe.

9. Um crocodilo raptou um bebê de sua mãe e prometeu devolvê-lo se a mãe respondesse corretamente “sim” ou “não” à questão : “Vou comer o bebê?”. O que a mãe respondeu e o que fez o crocodilo?

10. Um juiz determinou que uma testemunha respondesse “sim” ou “não” à questão “Sua próxima palavra será não?” Qual é a resposta da testemunha?

11. (Dilema do Enforcado) Os prisioneiros de um certo reino são sempre decapitados ou enforcados. Um prisioneiro conseguiu o privilégio de formular uma afirmação; se fosse falsa, ele seria enforcado, e se verdadeira, decapitado. Qual afirmação o prisioneiro poderia formular para não ser executado?

12. (Paradoxo de Protágoras) Um jovem advogado fez o seguinte trato com seu mestre, Protágoras: ele só pagaria pela sua instrução se conseguisse vencer o primeiro caso. Como ele nunca aceitava nenhum caso, Protágoras o acionou, e ele teve que se defender. Quem ganha a causa?

13. Construa um supermicrocomputador que seja:
    a) facílimo de carregar; b) econômico e simples de construir; c) infalível e universal; d) cujo sistema operacional seja tão simples que qualquer criança o opere.

(Sugestão: uma moeda, escrita "sim" de uma lado e "não" do outro. Faça qualquer pergunta à máquina, e em seguida uma nova pergunta apropriada.)

14. (Paradoxo do Livro sem Fim) Sobre uma mesa há um livro. Sua primeira página é bem espessa. A espessura da segunda é metade da da primeira, e assim por diante. O livro cumpre duas condições: primeiro, que cada página é seguida por uma sucessora cuja espessura é metade da anterior; e segundo, que cada página é separada da primeira por um número finito de páginas. Este livro tem última página?

15. (Paradoxo do Enforcado) Um juiz sentenciou um réu à morte pela forca, impondo a seguinte condição: que o réu seria enforcado de surpresa (isto é, sem poder saber em que dia), entre segunda e sexta feira da próxima semana, ao meio dia. Aconteceu a execução?

16. (Paradoxo do Ovo Inesperado) Imagine que você tem duas caixas à sua frente, numeradas de 1 a 2. Você vira as costas e um amigo esconde um ovo em uma delas. Ele pede que você as abra na ordem, e garante que você vai encontrar um ovo inesperado. É claro que o ovo não pode estar na caixa 2 (pois não seria inesperado), e portanto essa caixa está fora. Só pode estar na 1. Você vai abrindo as caixas e encontra o ovo na caixa 2. Ele estava certo, mas onde está seu erro de raciocínio?

17. (Paradoxo da Confirmação de Hempel) Suponha que um cientista queira provar que todo papagaio é verde. Essa afirmação equivale logicamente a "tudo que não é verde não é papagaio" e portanto todos os exemplos que confirmam a segunda sentença, confirmam a primeira. Um gato preto e branco não é verde, e não é papagaio, e portanto confirma que todo papagaio é verde. Onde está o erro?

18. Os seguintes paradoxos são devidos ao lógico Jean Buridanus (de seu livro *Sophismata*, do século XIV). Vamos admitir algumas hipóteses que parecem bastante razoáveis:

    - se sabemos alguma coisa, então acreditamos nisso;
    - se acreditamos que alguma coisa é verdadeira, então acreditamos nessa coisa;
    - se alguma coisa é falsa, então não pode fazer parte de nosso conhecimento.

Analise agora as sentenças abaixo, conhecidas como Paradoxos do conhecimento:

(a) "Ninguém acredita nesta sentença". Mostre que esta sentença não faz parte do conhecimento de ninguém.

(b) "Eu não acredito nesta sentença". É possível que você acredite nesta sentença?

(c) "Ninguém conhece esta sentença". Mostre que esta sentença é verdadeira, mas não faz parte do conhecimento de ninguém.

1.3.3 O que podemos aprender com os paradoxos?

Na seção precedente optamos por colocar os paradoxos como questões, desafiando você a tentar resolvê-los. Não mostramos as soluções, porque em geral elas não existem: os paradoxos não podem ser resolvidos como simples exercícios.

Na verdade os paradoxos colocam problemas que vão muito além da capacidade do conhecimento da lógica e mesmo da ciência. Portanto, não devemos nos surpreender com o fato de que os paradoxos possam conviver lado a lado com a lógica; o que podemos concluir é que o jardim organizado e seguro da lógica representa apenas uma parte da floresta selvagem do pensamento humano. Dentro deste pequeno jardim podemos usar nosso instrumento formal, que é o que será introduzido neste livro, e através dele colher algumas belas flores, algumas até surpreendentemente bonitas e curiosas; muitas outras podem estar perdidas na floresta, esperando ser descobertas. Dessas, este livro não vai tratar, mas esperamos que você pelo menos compreenda onde estão os limites do jardim da lógica.

Capítulo 2

Linguagem e Semântica da lógica proposicional clássica

2.1 Linguagens proposicionais

2.1.1 Assinaturas e linguagens

Apesar de diversos autores e obras tentarem apresentar a lógica como uma teoria do raciocínio, os paradoxos, como vimos, mostram que é difícil aceitar que raciocinemos com alguma lógica, pelo menos com uma determinada.

Preferimos considerar a lógica como uma teoria da *comunicação* do raciocínio, isto é, uma teoria da argumentação vista como o encadeamento de seqüências de *sentenças* por meio de uma (ou várias) relação do tipo "... segue de ...". Dessa forma, precisamos nos preocupar com três tarefas básicas:

1. especificar a *linguagem* com que expressamos os argumentos;
2. esclarecer os mecanismos que produzam ou que verifiquem os *argumentos válidos*; e
3. definir as noções de *provas* ou *demonstrações*, isto é, as seqüências de argumentos que produzam o fim desejado.

Contentamo-nos com as sentenças ditas *declarativas*, evitando assim sentenças interrogativas, temporais, modais, etc. De nosso ponto de vista, a lógica se interessa pelo *raciocínio matemático tradicional*, e para tal fim as sentenças declarativas são suficientes.

A rigor, podemos considerar também as sentenças *performativas*, usadas por exemplo em linguagens computacionais, que são também sentenças matemáticas, mas tais sentenças performativas podem ser interpretadas (i.e., reescritas) como sentenças declarativas.

Precisamos obter uma linguagem precisa para a matemática, que possa ela mesma ser *objeto* de análise matemática. Iniciamos nossa análise com as *proposições* estudando assim a *lógica sentencial*, também chamada de *cálculo proposicional*, *cálculo sentencial* ou *lógica proposicional clássica*, a qual denotaremos por LPC.

Mais tarde aumentaremos nossa linguagem para levar em conta as *propriedades* de indivíduos, expressas por meio de relações envolvendo indivíduos e variáveis, estudando então a *lógica de predicados* também chamada de *cálculo de predicados* ou *lógica de primeira ordem*.

De acordo com as tarefas básicas com que devemos nos ocupar, o processo de formalização da LPC consiste de:

1. especificar a linguagem formal;

2. especificar o processo de obter proposições *válidas* nessa linguagem;

3. especificar o processo de obter seqüências de proposições válidas.

Começamos especificando a linguagem, através da técnica de *definições indutivas*, que consiste em apresentar primeiro as proposições atômicas (i.e., que não são decomponíveis em sentenças mais simples) e, num segundo estágio, especificar como as sentenças mais complexas são construídas a partir dessas utilizando *conectivos*. Os conectivos são funções que conectam sentenças dadas para formar sentenças mais complexas. Exemplos de conectivos são a negação $\neg$, que permite formar a sentença $\neg\varphi$ a partir da sentença φ, a conjunção $\wedge$ que permite, dadas as sentenças φ e ψ, formar a sentença $(\varphi \wedge \psi)$, etc. Usualmente os conectivos são constantes (isto é, sem argumentos) ou funções de um ou dois argumentos. Porém, para não perder a generalidade, podemos definir linguagens formais onde podemos ter conectivos de qualquer número de argumentos. Chegamos assim à definição seguinte:

Definição 2.1.1. Uma *assinatura proposicional* é uma família $C = \{C_n\}_{n\in\mathbb{N}}$ tal que cada C_n é um conjunto, sendo que $C_n \cap C_m = \emptyset$ se $n \neq m$. Os elementos do conjunto C_n são chamados de *conectivos n-ários*. Em particular, os elementos de C_0 são chamados de *constantes*.

A idéia da construção de linguagens formais proposicionais é a seguinte: partindo de um conjunto infinito fixado $\mathcal{V} = \{p_n \ : \ n \in \mathbb{N}\}$ de símbolos, chamados de variáveis proposicionais, ou de fórmulas (ou proposições, ou sentenças) atômicas, construir as fórmulas mais complexas utilizando os conectivos de uma assinatura C, junto com alguns símbolos auxiliares (parênteses e vírgulas[1]). Essa linguagem dependerá da assinatura C (uma vez que o conjunto $\mathcal{V}$ e os símbolos auxiliares permanecerão fixados ao longo do estudo). Devemos destacar que, na maioria dos exemplos prácticos, uma assinatura C consiste de um número finito de conectivos. Nesses casos, existirá um $n \in \mathbb{N}$ tal que $C_m = \emptyset$ para todo $m \geq n$.

Consideraremos três símbolos auxiliares: '(' e ')' (parênteses esquerdo e direito, respectivamente) e ',' (vírgula). Seja Aux o conjunto formado pelos símbolos auxiliares. Assumimos que $\mathcal{V} \cap Aux = \emptyset$. Para evitar confusões, assumiremos também que as assinaturas C consideradas satisfazem: $C_n \cap \mathcal{V} = \emptyset = C_n \cap Aux$ para todo $n \in \mathbb{N}$. Dada uma assinatura C, e assumindo os conjuntos $\mathcal{V}$ e Aux, podemos considerar as *expressões* formadas com estes símbolos. Formalmente:

Definição 2.1.2. Seja C uma assinatura.

1. Definimos o conjunto $|C|$ como sendo

$$|C| = \bigcup\{C_n \ : \ n \in \mathbb{N}\}.$$

2. Seja $D_C = |C| \cup \mathcal{V} \cup Aux$. Uma *expressão* sobre C é uma seqüência finita de símbolos $\xi = s_1 \cdots s_n$ tal que $s_i \in D_C$ para $i = 1, \ldots, n$. O conjunto de todas as expressões sobre C é denotado por E_C.

Na definição acima, $\bigcup \mathcal{F}$ denota a união do conjunto de conjuntos $\mathcal{F}$, isto é,

$$\bigcup \mathcal{F} = \{x \ : \ x \in X \text{ para algum } X \in \mathcal{F}\}.$$

O objetivo da definição de linguagens formais é reconhecer (ou isolar), dentre as expressões possíveis sobre uma dada assinatura, certas expressões que "fazem sentido": as fórmulas. O conjunto das fórmulas constitui a linguagem gerada pela assinatura. Neste ponto, é útil estabelecer um paralelo com as linguagens naturais, em particular com a língua portuguesa. A partir de símbolos dados (as letras do alfabeto, com ou sem acentos e crases) podemos concatenar esses símbolos para construir expressões, tais como 'osspêpqtà' ou 'cerâmica'. Das duas expressões, apenas estaremos interessados na segunda, porque forma uma *palavra* (com sentido). E usando

[1] O uso destes símbolos auxiliares é apenas para fins de facilitar a individualização das componentes de uma fórmula, sendo, de fato, prescindíveis.

as palavras, estaremos interessados nas *frases* ou *sentenças declarativas*, isto é, seqüências finitas de palavras das quais *faz sentido* afirmar que são verdadeiras ou falsas, tais como 'Roma é a capital de Itália' ou 'dois mais dois é cinco'. Obviamente estas seqüências de palavras (*frases*) devem ser construídas seguindo certas regras gramaticais específicas (no caso, da língua portuguesa). As regras gramaticais das linguagens formais proposicionais são dadas apenas pela combinação das proposições básicas (atômicas) utilizando os conectivos dados de maneira iterada. Definimos então o seguinte:

Definição 2.1.3. Seja $C = \{C_n\}_{n\in\mathbb{N}}$ uma assinatura. A *linguagem gerada por C* é o conjunto $L(C)$ definido como sendo o menor dos subconjuntos X de E_C que satisfazem as seguintes propriedades:

1. $\mathcal{V} \subseteq X$;

2. se $n \in \mathbb{N}$, $c \in C_n$ e $\varphi_1, \ldots, \varphi_n \in X$ então $c(\varphi_1, \ldots, \varphi_n) \in X$ (em particular, $C_0 \subseteq X$).

Os elementos de $L(C)$ são chamados de *fórmulas* ou *sentenças* ou *proposições* sobre C. Da Definição 2.1.3 deduzimos o seguinte:

(i) $L(C)$ satisfaz as propriedades 1 e 2 da Definição 2.1.3;
(ii) Se $X \subseteq E_C$ satisfaz 1 e 2 da Definição 2.1.3 então $L(C) \subseteq X$.

A seguir exemplificaremos as definições introduzidas até agora, definindo três assinaturas especiais que utilizaremos ao longo deste livro para analisar a lógica proposicional clássica.

Exemplo 2.1.4. A assinatura C^0 consiste dos seguintes conectivos:

- $C^0_1 = \{\neg\}$ (*negação*);

- $C^0_2 = \{\vee\}$ (*disjunção*);

- $C^0_n = \emptyset$ se $n \neq 1$, $n \neq 2$.

Temos que $|C^0| = \{\neg, \vee\}$. As seqüências de símbolos $)) \vee \neg\neg,)p_1p_{345}(\vee$ e $\vee(p_1, p_2)$ são expressões de C^0 (apenas a segunda "faz sentido"). Porém a expressão $p_1p_3(\Rightarrow \spadesuit)$ não é uma expressão sobre C^0, porque contém os símbolos $\Rightarrow$ e $\spadesuit$ que não pertencem a D_{C^0}. Por outro lado, $\vee(p_1, \neg(p_2))$ e $\neg(\neg(\vee(p_{23}, p_1)))$ são fórmulas sobre C^0. O conjunto $L(C^0)$ das fórmulas sobre C^0 será chamado de Prop. Por simplicidade, escreveremos $\neg\varphi$ e $(\varphi \vee \psi)$

no lugar de $\neg(\varphi)$ e $\vee(\varphi, \psi)$, respectivamente. Freqüentemente economizaremos parênteses, omitindo os mais externos. Assim, escrevemos $\varphi \vee \psi$ ao invés de $(\varphi \vee \psi)$ quando a leitura não for comprometida. $\triangle$

Exemplo 2.1.5. A assinatura C^1 consiste dos seguintes conectivos:

- $C^1_1 = \{\neg\}$ (*negação*);
- $C^1_2 = \{\Rightarrow\}$ (*implicação*);
- $C^1_n = \emptyset$ se $n \neq 1$, $n \neq 2$.

Temos que $|C^1| = \{\neg, \Rightarrow\}$. É fácil ver que as expressões $\neg(\Rightarrow(p_3, p_7))$ e $\Rightarrow(p_0, \Rightarrow(\neg(p_0), p_1))$ são fórmulas sobre C^1. O conjunto $L(C^1)$ das fórmulas sobre C^1 será chamado de Prop_1. Como antes, escreveremos $\neg\varphi$ e $(\varphi \Rightarrow \psi)$ no lugar de $\neg(\varphi)$ e $\Rightarrow(\varphi, \psi)$, respectivamente. Também omitiremos freqüentemente parênteses, quando não houver risco de confusão. Assim, as duas fórmulas acima podem ser escritas como $\neg(p_3 \Rightarrow p_7)$ e $p_0 \Rightarrow (\neg p_0 \Rightarrow p_1)$. $\triangle$

Exemplo 2.1.6. A assinatura C^2 consiste dos seguintes conectivos:

- $C^2_1 = \{\neg\}$ *(negação);*
- $C^2_2 = \{\vee, \wedge, \Rightarrow\}$ *(disjunção, conjunção e implicação);*
- $C^2_n = \emptyset$ *se* $n \neq 1$, $n \neq 2$.

Temos que $|C^2| = \{\neg, \vee, \wedge, \Rightarrow\}$. Assumindo as mesmas convenções dos dois exemplos anteriores, as expressões $p_0 \Rightarrow p_0$, $p_4 \Rightarrow (p_4 \vee p_5)$ e $\neg p_6 \Rightarrow (p_3 \vee (p_3 \wedge p5))$ são fórmulas sobre C^2. O conjunto $L(C^2)$ das fórmulas sobre C^2 será chamado de Prop_2. $\triangle$

O leitor perspicaz poderia perguntar-se se é sempre possível *construir* o conjunto $L(C)$. De fato, a resposta é afirmativa. Assumindo a teoria de conjuntos (vide Apêndice XXX) como base formal para todas as nossas construções podemos simplesmente definir $L(C)$ através da seguinte fórmula:

$$L(C) = \bigcap\{X \subseteq E_C \ : \ X \text{ satisfaz as propriedades 1 e 2 da Definição 2.1.3}\}.$$

A notação $\bigcap \mathcal{F}$ acima utilizada denota a interseção do conjunto de conjuntos $\mathcal{F}$, isto é,

$$\bigcap \mathcal{F} = \{x \ : \ x \in X \text{ para todo } X \in \mathcal{F}\}.$$

Dado que o próprio E_C satisfaz as propriedades 1 e 2 da Definição 2.1.3, então o conjunto de *todos* os subconjuntos de E_C satisfazendo essas propriedades é não vazio, existindo portanto a sua interseção $L(C)$, a qual também satisfaz as propriedades 1 e 2. Claramente $L(C)$ é o menor dos subconjuntos de E_C satisfazendo estas propriedades.

Outra maneira (talvez mais construtiva) de provar a existência de $L(C)$ é a seguinte: Dado um conjunto arbitrário X, defina o conjunto

$$F_C(X) = \{c(\varphi_1, \dots, \varphi_n) \ : \ n \in \mathbb{N}, \ c \in C_n \ \text{ e } \ \varphi_1, \dots, \varphi_n \in X\} \cup X.$$

Defina agora, para cada $n \in \mathbb{N}$ e para cada conjunto X, o seguinte conjunto: $F_C^0(X) = X$, e $F_C^{n+1}(X) = F_C(F_C^n(X))$. Claramente este processo sempre pode ser (idealmente) efetuado. Finalmente, juntando todas as expressões definidas através deste processo, teremos uma construção do conjunto $L(C)$:

$$L(C) = \bigcup \{F_C^n(\mathcal{V}) \ : \ n \in \mathbb{N}\}.$$

Deixamos para o leitor provar que esta última construção define de fato o conjunto $L(C)$.

Observação 2.1.7. As letras $\varphi_1, \dots, \varphi_n$ utilizadas para falar em fórmulas arbitrárias, assim como a letra c usada para denotar um conectivo arbitrário, são chamadas *metavariáveis*, isto é, objetos na *metalinguagem* que se referem aos objetos sendo estudados. A metalinguagem é simplesmente a linguagem (natural ou matemática) na qual *falamos sobre* nossa linguagem formal. Usaremos letras gregas minúsculas (com ou sem índices) como meta-variáveis de fórmulas. Reservamos $p_0, p_1, p_2, \dots$ para denotar as variáveis proposicionais propriamente ditas.

2.1.2 Indução estrutural

Em geral, as *propriedades* das proposições são estabelecidas por um *processo indutivo* (similar ao usado na definição de $L(C)$) pelo qual, se queremos *provar* uma certa propriedade P a respeito das proposições, começamos mostrando que P vale para as proposições atômicas, e depois, supondo a propriedade válida para as componentes de uma proposição, provamo-la para a proposição mais complexa.

A rigor, tal processo pode ser inteiramente convertido numa prova *aritmética*, usando a conhecida *indução aritmética* com relação a um parâmetro numérico qualquer. Por exemplo podemos usar como parâmetro da indução o número de símbolos (i.e., o comprimento da proposição), o número de parênteses, etc.

O teorema a seguir demonstra que de fato as propriedades das proposições podem ser provadas pelo processo indutivo. Este teorema é muito importante porque muitas propriedades da lógica (na verdade, poderíamos dizer que a maioria) são provadas através deste processo indutivo.

Introduzimos a seguinte notação (semi)formal: $P(\varphi)$ denota que a sentença φ satisfaz a propriedade P (note que $P(\varphi)$ é um enunciado da metalinguagem, que pode portanto ser verdadeiro ou falso). Por exemplo, $P(\varphi)$ poderia ser o enunciado "a sentença φ tem tantos parênteses esquerdos quantos parênteses direitos", ou o enunciado "em φ somente ocorrem variáveis proposicionais e conectivos de aridade 1 ou 2". Temos portanto o seguinte:

Teorema 2.1.8. *Seja P uma propriedade sobre expressões, e considere o conjunto*

$$X = \{\varphi \in L(C) \ : \ P(\varphi) \text{ vale}\}.$$

Suponha que X satisfaz o seguinte:

1. *$\mathcal{V} \subseteq X$;*

2. *para todo $n \in \mathbb{N}$, $c \in C_n$ e $\varphi_1, \ldots, \varphi_n \in L(C)$, se $\varphi_i \in X$ (para $i = 1, \ldots, n$) então $c(\varphi_1, \ldots, \varphi_n) \in X$.*

Então concluímos que $X = L(C)$.

Demonstração: Por definição temos que $X \subseteq L(C)$. Uma vez que $L(C)$ é o menor conjunto de expressões sobre C com as propriedades 1 e 2 satisfeitas por X (veja Definição 2.1.3) inferimos que $L(C) \subseteq X$, e daí $X = L(C)$. $\square$

Quando este teorema é aplicado para provar alguma propriedade de proposições, dizemos que se trata de uma prova por *indução nas proposições*.

Observe que uma formulação equivalente do Teorema 2.1.8 é a seguinte:

Teorema 2.1.9. *Seja P uma propriedade. Suponha o seguinte:*

1. *$P(p)$ vale para toda proposição atômica p;*

2. *para todo $n \in \mathbb{N}$, $c \in C_n$ e $\varphi_1, \ldots, \varphi_n \in L(C)$, se $P(\varphi_i)$ vale (para $i = 1, \ldots, n$) então $P(c(\varphi_1, \ldots, \varphi_n))$ vale.*

Então concluímos que $P(\varphi)$ vale para toda sentença φ de $L(C)$.

Veremos a seguir algumas aplicações do Teorema 2.1.8 que nos ajudarão a entender as características das linguagens $L(C)$. Um resultado importante a ser provado é com relação à unicidade da escrita das fórmulas: se

$c(\varphi_1, \ldots, \varphi_n) = d(\psi_1, \ldots, \psi_m)$ então $c = d$ (logo $n = m$) e $\varphi_i = \psi_i$ para todo $i = 1, \ldots, n$. Para isso precisamos provar alguns resultados prévios:

Proposição 2.1.10. *Para toda fórmula $\varphi \in L(C)$, o número de parênteses esquerdos '(' é o mesmo que o número de parênteses direitos ')'.*

Demonstração: Seja $pe(\varphi)$ o número de parênteses esquerdos da fórmula φ, e seja $pd(\varphi)$ o número de parênteses direitos da fórmula φ. Observe que os valores $pe(\varphi)$ e $pd(\varphi)$ de φ sempre podem ser calculados, pois trata-se apenas de contar certos símbolos numa seqüência finita de símbolos. Considere agora o conjunto

$$X = \{\varphi \in L(C) \ : \ pe(\varphi) = pd(\varphi)\}.$$

Claramente $\mathcal{V} \subseteq X$, pois $pe(p) = pd(p) = 0$ para toda $p \in \mathcal{V}$. Dados $\varphi_1, \ldots, \varphi_n \in X$ e $c \in C_n$ então $pe(\varphi_i) = pd(\varphi_i)$ para todo $i = 1, \ldots, n$. Logo

$$pe(c(\varphi_1, \ldots, \varphi_n)) = 1 + \sum_{i=1}^{n} pe(\varphi_i) = 1 + \sum_{i=1}^{n} pd(\varphi_i) = pd(c(\varphi_1, \ldots, \varphi_n)),$$

portanto $c(\varphi_1, \ldots, \varphi_n) \in X$. Pelo Teorema 2.1.8 obtemos que $X = L(C)$. $\square$

Note que na prova acima não utilizamos a unicidade da escrita das fórmulas: na equação $pe(c(\varphi_1, \ldots, \varphi_n)) = 1 + \sum_{i=1}^{n} pe(\varphi_i)$ estamos apenas utilizando *uma* maneira possível de calcular o valor $pe(\varphi)$ para a fórmula $\varphi = c(\varphi_1, \ldots, \varphi_n)$, independentemente da maneira específica de construir φ.

Dada uma expressão $\xi = s_1 \cdots s_n$ sobre C (veja Definição 2.1.2), definimos um *segmento inicial de* ξ como sendo uma expressão $\xi' = s_1 \cdots s_m$, com $m < n$.

Assim, $p_3(\vee p_2$ e $p_3($ são segmentos iniciais de $p_3(\vee p_2((p_4$. O caso interessante é quando consideramos segmentos iniciais de fórmulas: eles deixam de ser fórmulas, como mostra o seguinte resultado técnico.

Lema 2.1.11. *Seja φ uma fórmula e ξ um segmento inicial de φ. Então $\xi \notin L(C)$.*

A demonstração deste lema é imediata a partir da Proposição 2.1.10, e é deixada como exercício para o leitor. Finalmente podemos provar a unicidade da escrita das fórmulas:

Proposição 2.1.12. *Sejam $c \in C_n$, $d \in C_m$, e $\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m \in L(C)$. Se $c(\varphi_1, \ldots, \varphi_n) = d(\psi_1, \ldots, \psi_m)$ então $c = d$, $n = m$ e $\varphi_i = \psi_i$ para $i = 1, \ldots, n$.*

Demonstração: De $c(\varphi_1, \ldots, \varphi_n) = d(\psi_1, \ldots, \psi_m)$ (igualdade como seqüências finitas de símbolos) inferimos imediatamente que $c = d$, portanto $n = m$ (lembre que $C_n \cap C_m = \emptyset$ se $n \neq m$). Considere agora as seqüências finitas de símbolos φ_1 e ψ_1. Se φ_1 e ψ_1 têm o mesmo comprimento então elas coincidem, pois $c(\varphi_1, \ldots, \varphi_n) = d(\psi_1, \ldots, \psi_m)$. Caso contrário, uma das fórmulas deve ser um segmento inicial da outra. Mas isto é impossível, por causa do Lema 2.1.11. Daqui inferimos que as duas fórmulas têm o mesmo comprimento e então $\varphi_1 = \psi_1$. O mesmo argumento pode ser usado para provar que $\varphi_i = \psi_i$ para $i = 2, \ldots, n$. $\square$

Corolário 2.1.13. *Seja $\varphi \in L(C)$. Então vale uma e apenas uma das seguintes afirmações:*

1. $\varphi \in \mathcal{V} \cup C_0$;
2. *existe um único $n \geq 1$, um único $c \in C_n$ e únicas $\varphi_1, \ldots, \varphi_n \in L(C)$ tais que $\varphi = c(\varphi_1, \ldots, \varphi_n)$.*

O Corolário 2.1.13 será utilizado daqui em diante sem menção explícita. Podemos também por um processo indutivo (conhecido como *definição por recursão*) definir propriedades quaisquer a respeito de proposições. Em particular, temos um método para definir funções com domínio $L(C)$.

Notação 2.1.14. A partir de agora usaremos a seguinte notação usual: uma função f do conjunto X no conjunto Y será denotada por $f : X \to Y$, e dado um conjunto X e um número natural n, então X^n denotará o conjunto de todas as n-uplas formadas por elementos de X. Por convenção, X^0 é o conjunto unitário $\{*\}$.

Teorema 2.1.15. *Seja S um conjunto. Suponha que, para cada $n \in \mathbb{N}$ e para $c \in C_n$, existe uma função $H_c : S^n \to S$ (em particular, existe um elemento H_c de S para cada constante $c \in C_0$). Seja $H : \mathcal{V} \to S$ uma função. Então existe uma única função $f : L(C) \to S$ satisfazendo o seguinte:*

1. *$f(p) = H(p)$ para toda $p \in \mathcal{V}$;*
2. *$f(c(\varphi_1, \ldots, \varphi_n)) = H_c(f(\varphi_1), \ldots, f(\varphi_n))$ para todo $n \in \mathbb{N}$, $c \in C_n$ e $\varphi_1, \ldots, \varphi_n \in L(C)$. Em particular, $f(c) = H_c$ se $c \in C_0$.*

Demonstração: Considere o seguinte conjunto de sentenças:

$$X = \{\varphi \in L(C) \ : \ \text{ existe um único } s \in S \text{ tal que } f(\varphi) = s\}.$$

Claramente $\mathcal{V} \subseteq X$, pois a primeira cláusula (e apenas ela) se aplica a cada variável $p \in \mathcal{V}$. Seja $c \in C_n$ e $\varphi_1, \ldots, \varphi_n \in X$. Dado que existe um único valor $s_i = f(\varphi_i)$ para cada $i = 1, \ldots, n$, e dado que $c(\varphi_1, \ldots, \varphi_n)$ admite uma escrita única (pela Proposição 2.1.12) então existe um único valor designado para $c(\varphi_1, \ldots, \varphi_n)$ por f, dado por

$$f(c(\varphi_1, \ldots, \varphi_n)) = H_c(s_1, \ldots, s_n).$$

Portanto $c(\varphi_1, \ldots, \varphi_n) \in X$. Pelo Teorema 2.1.8 temos que $X = L(C)$. Isto prova que existe *ao menos uma* função $f : L(C) \to S$ satisfazendo as propriedades do enunciado do teorema. Provaremos agora que a função f é de fato única. Para isso, suponhamos que $g : L(C) \to S$ é uma função que satisfaz as propriedades do enunciado. Considere o conjunto

$$Y = \{\varphi \in L(C) \ : \ f(\varphi) = g(\varphi)\}.$$

É imediato que, pelo Teorema 2.1.8, $Y = L(C)$ (deixamos os detalhes da prova para o leitor), portanto $g = f$. Isto conclui a prova do teorema. □

2.1.3 A linguagem da LPC

Os resultados apresentados até agora são gerais e valem para qualquer linguagem da forma $L(C)$. Vamos agora nos concentrar em três assinaturas particulares, que usaremos daqui em diante, sendo que cada uma delas é suficiente para descrever a lógica proposicional clássica (de aqui em diante, abreviada por LPC). As assinaturas são C^0, C^1 e C^2, descritas nos exemplos 2.1.4, 2.1.5 e 2.1.6, respectivamente.

Como um exemplo de aplicação do Teorema 2.1.15, definimos a função *grau* de uma proposição. Considere $Max(X)$ como sendo o máximo do conjunto X (com relação a uma ordem dada; veja o Capítulo 5). Então definimos o seguinte:

Definição 2.1.16. A função *grau* é a função $g : \mathsf{Prop} \to \mathbb{N}$ definida como segue:

1. $g(\varphi) = 0$, se φ é atômica;
2. $g(\varphi \vee \psi) = Max(\{g(\varphi), g(\psi)\}) + 1$;

3. $g(\neg\varphi) = g(\varphi) + 1$.

A partir do Teorema 2.1.15 é imediato que a função g está bem definida, considerando $S = \mathbb{N}$, $H(p) = 0$ para todo $p \in \mathcal{V}$, $H_\neg(x) = x + 1$ para todo $x \in \mathbb{N}$, e $H_\vee(x, y) = Max(\{x, y\}) + 1$.

Analogamente, podemos definir outra medida para as fórmulas.

Definição 2.1.17. A função *complexidade* é a função $l : \mathsf{Prop} \to \mathbb{N}$ definida como segue:

1. $l(\varphi) = 1$, se φ é atômica;
2. $l(\varphi \vee \psi) = l(\varphi) + l(\psi) + 1$;
3. $l(\neg\varphi) = l(\varphi) + 1$.

De novo, a partir do Teorema 2.1.15 prova-se imediatamente que a função l está bem definida, considerando $S = \mathbb{N}$, $H(p) = 1$ para todo $p \in \mathcal{V}$, $H_\neg(x) = x + 1$ para todo $x \in \mathbb{N}$, e $H_\vee(x, y) = x + y + 1$.

Observação 2.1.18. Observe que as funções g e l podem ser definidas em qualquer linguagem $L(C)$: para g definimos

1. $g(\varphi) = 0$, se $\varphi \in \mathcal{V} \cup C_0$;
2. $g(c(\varphi_1, \ldots, \varphi_n)) = Max(\{g(\varphi_1), \ldots, g(\varphi_n)\}) + 1$.

Para l definimos

1. $l(\varphi) = 1$, se $\varphi \in \mathcal{V} \cup C_0$;
2. $l(c(\varphi_1, \ldots, \varphi_n)) = 1 + \sum_{i=1}^{n} l(\varphi_i)$.

Outra função útil a ser definida sobre o conjunto Prop é a função *Var*, que associa a cada fórmula φ o conjunto das variáveis proposicionais que ocorrem na fórmula. Usaremos o símbolo $\wp(X)$ para denotar o conjunto das partes do conjunto X, isto é,

$$\wp(X) = \{Y \ : Y \subseteq X\}.$$

Definição 2.1.19. Definimos a função $Var : \mathsf{Prop} \to \wp(\mathcal{V})$ como segue:

1. $Var(p) = \{p\}$ se $p \in \mathcal{V}$;
2. $Var(\varphi \vee \psi) = Var(\varphi) \cup Var(\psi)$;

3. $Var(\neg\varphi) = Var(\varphi)$.

Claramente a função *Var* está bem definida, usando o Teorema 2.1.15. Basta tomar $S = \wp(\mathcal{V})$, $H(p) = \{p\}$ para todo $p \in \mathcal{V}$, $H_\neg(X) = X$ e $H_\vee(X, Y) = X \cup Y$ para todo $X, Y \subseteq \mathcal{V}$.

Observação 2.1.20. Novamente a função *Var* pode ser definida em toda linguagem $L(C)$, substituindo a segunda e a terceira cláusulas da definição acima pela mais geral:

$$Var(c(\varphi_1, \ldots, \varphi_n)) = \bigcup_{i=1}^{n} Var(\varphi_i).$$

Em particular, $Var(c) = \emptyset$ se $c \in C_0$.

A noção de *subfórmula* de uma fórmula também pode ser definida indutivamente:

Definição 2.1.21. Definimos a função *conjunto de subfórmulas* como sendo a função $SF : \mathsf{Prop} \to \wp(\mathsf{Prop})$ tal que:

1. $SF(\varphi) = \{\varphi\}$, se φ é atômica;
2. $SF(\varphi \vee \psi) = SF(\varphi) \cup SF(\psi) \cup \{\varphi \vee \psi\}$;
3. $SF(\neg\varphi) = SF(\varphi) \cup \{\neg\varphi\}$.

É possível provar, por indução na complexidade da fórmula φ, que $SF(\varphi)$ está bem definida (deixamos como exercício para o leitor). Os elementos de $SF(\varphi)$ são fórmulas, chamadas *subfórmulas de* φ. A partir desta função podemos definir a função *conjunto de subfórmulas estritas* da maneira seguinte:

$$SFE : \mathsf{Prop} \to \wp(\mathsf{Prop}), \quad SFE(\varphi) = SF(\varphi) - \{\varphi\}$$

(a diferença entre o conjunto X e o conjunto Y será denotada por $X - Y$). Os elementos de $SFE(\varphi)$ são chamados de *subfórmulas estritas de* φ.

Novamente, as funções *SF* e *SFE* existem em toda linguagem $L(C)$, portanto poderemos falar em subfórmula e em subfórmula estrita de uma dada fórmula em qualquer linguagem.

Exemplo 2.1.22. p_2, p_3, p_{10}, $\neg p_2$, $\neg p_3$, $(p_{10} \vee \neg p_3)$ e $(p_{10} \vee \neg p_3) \vee \neg p_2$ são todas as subfórmulas de $(p_{10} \vee \neg p_3) \vee \neg p_2$. $\triangle$

Apresentamos finalmente um resultado adicional que fundamenta as definições por recursão. A definição a seguir é uma maneira de explicitar indutivamente os passos na construção de uma proposição (ou sentença, ou fórmula):

Definição 2.1.23. Seja ξ uma expressão sobre a assinatura C^0. Uma seqüência finita de expressões $\xi_1 \dots \xi_n$ é chamada *seqüência de formação* de ξ se $\xi_n = \xi$ e, para cada $1 \leq i \leq n$:

1. ξ_i é uma variável proposicional, ou
2. existem $j, k < i$ tais que $\xi_i = \xi_j \vee \xi_k$, ou
3. existe $j < i$ tal que $\xi_i = \neg \xi_j$.

Dizemos que ξ é uma *fórmula bem-formada* se possui uma seqüência de formação.

Teorema 2.1.24. Prop *é o conjunto de todas as fórmulas bem-formadas.*

Demonstração: Seja S o conjunto de todas as fórmulas bem-formadas. Dado que $S \subseteq E_{C^0}$ satisfaz 1 e 2 da Definição 2.1.3 então $\mathsf{Prop} \subseteq S$

Por outro lado, suponha que ξ tem seqüência de formação $\xi_1 \dots \xi_n = \xi$. Vamos mostrar por indução em n que $\xi \in \mathsf{Prop}$.
(i) $n = 1$. Nesse caso, $\xi_1 = \xi$, e ξ_1 é forçosamente uma fórmula atômica, logo $\xi \in \mathsf{Prop}$.
(ii) Suponha que para $m < n$, o resultado vale, isto é, toda expressão que admite uma seqüência de formação com comprimento $m < n$ está em Prop, para um dado $n \geq 1$.

Nesse caso, $\xi_n = \xi_i \vee \xi_j$ (ou $\xi_n = \neg \xi_i$), $i, j < n$. Então ξ_i, $\xi_j \in \mathsf{Prop}$, e por construção de Prop, $\xi_n \in \mathsf{Prop}$. □

Claramente, a Definição 2.1.23 pode ser dada em qualquer linguagem $L(C)$, e o Teorema 2.1.24 também vale em todo $L(C)$.

Exemplos 2.1.25.

1. $p_1\, p_2\, p_3\, \neg p_3\, (p_1 \vee p_2)\, (\neg p_3 \vee (p_1 \vee p_2))$ é uma seqüência de formação de $(\neg p_3 \vee (p_1 \vee p_2))$.
2. $p_2\, p_3\, p_4\, (p_2 \vee p_3)\, \neg p_3$ é uma seqüência de formação de $\neg p_3$.
3. $p_3\, \neg p_3$ é outra seqüência de formação de $\neg p_3$ (esta direta, a outra indireta).

△

2.1.4 Exercícios

1. Seja C^2 a asinatura definida no Exemplo 2.1.6. Para cada conjunto finito X, denotamos o seu número de elementos por $\overline{X}$. Considere as funções l, nc e Var definidas como segue:

   - $l(\varphi)$ = complexidade da fórmula φ (como na Observação 2.1.18);
   - $nc(\varphi)$ = número de conectivos que ocorrem na fórmula φ;
   - $Var(\varphi)$ = conjunto das variáveis proposicionais que ocorrem em φ (como na Observação 2.1.20).

   Provar que si não ocorrem variáveis proposicionais repetidas em φ, então $l(\varphi) = nc(\varphi) + \overline{Var(\varphi)}$.

2. Considere novamente a assinatura C^2. Provar o seguinte:

   (a) As expressões '$\vee(p_4, \neg(\wedge(p_4, p_0)))$', '$\Rightarrow (\neg(\neg(\neg(p_5))), \Rightarrow (p_2, p_2))$' e '$\neg(\Rightarrow (\vee(\neg(p_4), p_3), \wedge(p_1, p_0)))$' são fórmulas bem-formadas.

   (b) As expressões '$p_1()() \Rightarrow$', '$\Rightarrow (p_1, p_1, \neg(p_2))$' e '$\vee(p_4, \neg(\wedge(p_4, p_0))$' não são fórmulas bem-formadas.

3. Provar o Lema 2.1.11: Se $\varphi \in L(C)$ e ξ é segmento inicial de φ então $\xi \notin L(C)$.

4. Provar o seguinte: Se $\varphi \in L(C)$ e ξ é segmento inicial de φ de comprimento ≥ 2 então $pe(\xi) > pd(\xi)$. (Aqui, estamos estendendo as funções pe e pd que contam os parênteses esquerdos e direitos, respetivamente, para o conjunto E_C das expressões sobre C.)

5. Completar a prova do Teorema 2.1.15.

6. Provar por indução na complexidade de $\varphi \in L(C)$ que $SF(\varphi)$ sempre pode ser calculado em qualquer assinatura C (generalizando a Definição 2.1.21).

7. Generalize a Definição 2.1.23 e prove o análogo do Teorema 2.1.24 para uma linguagem $L(C)$ arbitrária.

2.2 Semântica da LPC

2.2.1 Semântica dos conectivos

Na definição da linguagem da LPC (isto é, na definição do conjunto Prop) tínhamos como conectivos lógicos apenas os operadores $\vee$ e $\neg$. Observe que estes conectivos não têm nenhum significado: são apenas símbolos, portanto eles não têm o sentido usual de "ou" e "não". Para que isso aconteça, interpretaremos os conectivos $\vee$ e $\neg$ utilizando funções (chamadas de *funções de verdade*), que envolvem os *valores de verdade* 1 ("verdadeiro") e 0 ("falso"). Seja $\mathbf{2}$ o conjunto $\mathbf{2} = \{0, 1\}$ dos valores de verdade. Definimos funções $- : \mathbf{2} \to \mathbf{2}$ e $\sqcup : \mathbf{2}^2 \to \mathbf{2}$ através das seguintes tabelas (adotaremos eventualmente as mesmas convenções de notação infixa da Seção 2.1 com relação aos operadores binários, escrevendo o operador *entre* os argumentos):

p	q	$p \sqcup q$
1	1	1
1	0	1
0	1	1
0	0	0

p	$-p$
1	0
0	1

A partir daí, podemos definir indutivamente o valor de verdade 1 ou 0 de uma proposição a partir do valor de suas componentes atômicas. Formalmente, definimos uma função *valoração* $v : \mathsf{Prop} \to \mathbf{2}$ como segue:

Definição 2.2.1. Uma função $v : \mathsf{Prop} \to \mathbf{2}$ é uma *valoração* (clássica) se satisfaz o seguinte:

1. $v(\varphi \vee \psi) = \sqcup(v(\varphi), v(\psi))$;
2. $v(\neg\varphi) = -(v(\varphi))$.

Observação 2.2.2. Note que sempre é possível definir valorações, por causa do Teorema 2.1.15. Com efeito, basta considerar uma função $v_0 : \mathcal{V} \to \mathbf{2}$ arbitrária. Usando o Teorema 2.1.15 e as funções $\vee$ e $\neg$ definidas pelas tabelas-verdade acima, temos que existe uma única valoração $v : \mathsf{Prop} \to \mathbf{2}$ estendendo v_0, isto é, tal que $v(p) = v_0(p)$ para toda variável proposicional p. A partir desta observação vemos que uma valoração é determinada pelos valores que ela toma nas variáveis proposicionais, isto é:

Proposição 2.2.3. *Se v e v' são duas valorações que coincidem em $\mathcal{V}$ (ou seja, tais que $v(p) = v'(p)$ para toda $p \in \mathcal{V}$) então $v(\varphi) = v'(\varphi)$ para toda $\varphi \in \mathsf{Prop}$, portanto $v = v'$.*

Demonstração: Imediata, usando o Teorema 2.1.15. $\square$

Graças à Proposição 2.2.3 vemos que, para definir uma valoração, basta dar uma função $v_0 : \mathcal{V} \to \mathbf{2}$. Por outro lado, podemos refinar este resultado, mostrando que o valor de verdade $v(\varphi)$ dado a uma fórmula φ por uma valoração v depende apenas dos valores de verdade $v(p)$ das variáveis p que ocorrem em φ.

Proposição 2.2.4. *Seja φ uma fórmula. Se v e v' são duas valorações tais que $v(p) = v'(p)$ para toda $p \in \mathrm{Var}(\varphi)$ então $v(\varphi) = v'(\varphi)$.*

Demonstração: Por indução na complexidade $n = l(\varphi)$.

Se $n = 1$ então $\varphi \in \mathcal{V}$ e o resultado vale trivialmente. Suponha que, dado $n \geq 1$, o resultado vale para toda sentença φ tal que $l(\varphi) \leq n$, e seja φ tal que $l(\varphi) = n + 1$. Temos dois casos para analisar:

Caso 1: $\varphi = \neg\psi$. Dado que $Var(\psi) = Var(\varphi)$ então v e v' coincidem em todas as variáveis de ψ, sendo que $l(\psi) = n$. Portanto, $v(\psi) = v'(\psi)$, donde $v(\neg\psi) = -(v(\psi)) = -(v'(\psi)) = v'(\neg\psi)$.

Caso 2: $\varphi = (\psi \vee \gamma)$. Dado que $Var(\psi) \subseteq Var(\varphi)$ então v e v' coincidem em todas as variáveis de ψ, sendo que $l(\psi) \leq n$. Portanto, $v(\psi) = v'(\psi)$. Analogamente provamos que $v(\gamma) = v'(\gamma)$ e então $v(\psi \vee \gamma) = \sqcup(v(\psi), v(\gamma)) = \sqcup(v'(\psi), v'(\gamma)) = v'(\psi \vee \gamma)$. Isto conclui a demonstração. $\square$

Observação 2.2.5. Podemos caracterizar as valorações da maneira seguinte: uma valoração é uma função $v : \mathsf{Prop} \to \mathbf{2}$ satisfazendo o seguinte:

1. $v(\varphi \vee \psi) = \begin{cases} 1 & \text{se } \ v(\varphi) = 1 \ \text{ ou } \ v(\psi) = 1 \\ 0 & \text{caso contrário} \end{cases}$

2. $v(\neg\varphi) = \begin{cases} 1 & \text{se } \ v(\varphi) = 0 \\ 0 & \text{se } \ v(\varphi) = 1 \end{cases}$

As definições semânticas (isto é, através de tabelas-verdade) da negação e da disjunção correspondem com as nossas intuições: dadas duas proposições P e Q (na linguagem natural), então a proposição composta "P ou Q" deve ser verdadeira se uma delas (ou as duas) são verdadeiras, e deve ser falsa se as duas componentes P e Q são falsas. Isto é descrito pela função de verdade $\sqcup$ definida acima. Por outro lado, a proposição "não P" (ou "não é o caso que P") é verdadeira se P é falsa, e vice-versa. Este comportamento é representado pela tabela da função $-$ definida acima.

É claro que existem outros conectivos na linguagem natural que podem ser modelados (de maneira simplista) através de tabelas-verdade. Os conectivos que analisaremos semanticamente a seguir são os que correspondem com a assinatura proposicional C^2 (veja Exemplo 2.1.6).

A *conjunção* $\sqcap : \mathbf{2}^2 \to \mathbf{2}$ e a *implicação material* $\sqsupset : \mathbf{2}^2 \to \mathbf{2}$ são definidas através das seguintes tabelas-verdade:

p	q	$p \sqcap q$
1	1	1
1	0	0
0	1	0
0	0	0

p	q	$p \sqsupset q$
1	1	1
1	0	0
0	1	1
0	0	1

Observe que a tabela da conjunção coincide com as nossas intuições com relação à conjunção de duas proposições: a conjunção "P e Q" é verdadeira apenas no caso em que ambas componentes, P e Q, o são. Por outro lado, a implicação material é mais complicada de justificar. Em princípio, pode-se pensar que a noção de *implicação* esta relacionada com a noção de causa-efeito: dizer que "P implica Q" sugere que P é uma causa para Q. Esta leitura surge, por exemplo, quando são enunciadas leis ou regras da forma "Se P, então Q" (por exemplo, na física). Esta leitura pressupõe portanto uma relação entre os antecedente P e o consequente Q. A interpretação que é feita em lógica clássica da implicação é diferente: trata-se apenas da *preservação da verdade*. Assim, afirmar que "P implica Q" nada mais diz do que a verdade de P garante a verdade de Q. Para ser mais explícito, uma implicação é verdadeira se, toda vez que o antecedente é verdadeiro, então o consequente também o é. Assim, uma frase do tipo 'Roma é a capital de Itália implica que Brasil está situado na América do sul' deve ser considerada verdadeira, do ponto de vista da lógica clássica (e no atual estado das coisas), dado que a verdade do antecedente é mantida no consequente. Em outras palavras, uma implicação material "P implica Q" é falsa (numa dada situação) quando o antecedente P é verdadeiro mas o consequente Q é falso. Nessa situação, não é o caso que a verdade de P foi preservada por Q. A definição de implicação impõe que um antecedente falso implique qualquer proposição (verdadeira ou falsa), uma vez que não é o caso que o antecedente foi verdadeiro e o consequente foi falso. Isto tem como conseqüência que, partindo de premissas falsas, não é mais possível raciocinar utilizando lógica clássica, pois qualquer conclusão pode ser tirada a partir daí. Existem muitas outras lógicas (chamadas de *não-clássicas*) desenvolvidas principalmente a partir do século XX, em que são definidas noções de implicação diferentes da implicação material, por exemplo, as *implicações relevantes* e as *implicações contrafatuais*.

Existem outros dois conectivos binários *clássicos* dignos de mencionar: a *disjunção exclusiva* $\veebar : \mathbf{2}^2 \to \mathbf{2}$ e o *bicondicional* $\Leftrightarrow: \mathbf{2}^2 \to \mathbf{2}$. As tabelas destes conectivos são as seguintes:

p	q	$p \veebar q$
1	1	0
1	0	1
0	1	1
0	0	0

p	q	$p \Leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

A idéia por trás destes conectivos é a seguinte: $\veebar$ representa uma disjunção exclusiva, isto é, um tipo de disjunção "P ou Q" da linguagem natural que é verdadeira se *exatamente* uma das proposições P, Q é verdadeira. Por exemplo, o sentido esperado da frase "Iremos ao cinema ou ao teatro" é o de ser verdadeiro quando formos ao cinema ou ao teatro, mas não ambas coisas. Por outro lado, o bicondicional reflete que as sentenças envolvidas são equivalentes, no sentido que elas têm o mesmo valor de verdade: ou ambas são verdadeiras, ou ambas são falsas. Estes conectivos não foram incluídos na assinatura C^2 porque eles podem ser expressos em termos dos outros conectivos. Com efeito, a função $\veebar(p,q)$ pode ser calculada pela função $f(p,q) = \sqcap(\sqcup(p,q), -(\sqcap(p,q)))$, enquanto que $\Leftrightarrow(p,q)$ pode ser calculada como $g(p,q) = \sqcap(\sqsupset(p,q), \sqsupset(q,p))$:

p	q	$p \sqcup q$	$p \sqcap q$	$-(p \sqcap q)$	$f(p,q)$
1	1	1	1	0	0
1	0	1	0	1	1
0	1	1	0	1	1
0	0	0	0	1	0

p	q	$p \sqsupset q$	$q \sqsupset p$	$g(p,q)$
1	1	1	1	1
1	0	0	1	0
0	1	1	0	0
0	0	1	1	1

Note que $f(p,q)$ expressa que alguma das sentenças p, q deve ser verdadeira, mas não podem ser ambas verdadeiras. A função $g(p,q)$ expressa que se p é verdadeiro então q também o é, e vice-versa.

O fenómeno de poder expressar certas funções de verdade em termos de outras é frequente: assim, a implicação material e a conjunção podem ser expressas em termos da disjunção e da negação como segue: $h(p,q) = \sqcup(-(p), q)$ e $k(p,q) = -(\sqcup(-(p), -(q)))$, respectivamente.

p	q	$-p$	$-p \sqcup q$
1	1	0	1
1	0	0	0
0	1	1	1
0	0	1	1

p	q	$-p$	$-q$	$-p \sqcup -q$	$-(-p \sqcup -q)$
1	1	0	0	0	1
1	0	0	1	1	0
0	1	1	0	1	0
0	0	1	1	1	0

O fato de poder expressar as funções $\sqsupset$ e $\sqcap$ (e, a posteriori, as funções $\veebar$ e $\Leftrightarrow$) em termos das funções $\sqcup$ e $-$ justifica a escolha da assinatura C^0 para representar a lógica proposicional clássica (LPC). Voltaremos depois (na Subseção 2.2.4) sobre a questão da definibilidade de funções de verdade em termos de outras funções.

Finalmente, observemos que podemos estender (ou modificar) a Definição 2.2.1 de valoração para as asinaturas C^2 e C^1:

Definição 2.2.6. (i) Uma função $v : \mathsf{Prop}_1 \to \mathbf{2}$ é uma *valoração* se satisfaz o seguinte:

1. $v(\varphi \Rightarrow \psi) = \sqsupset(v(\varphi), v(\psi))$;
2. $v(\neg\varphi) = -(v(\varphi))$.

(ii) Uma função $v : \mathsf{Prop}_2 \to \mathbf{2}$ é uma *valoração* se satisfaz o seguinte:

1. $v(\varphi \vee \psi) = \sqcup(v(\varphi), v(\psi))$;
2. $v(\varphi \wedge \psi) = \sqcap(v(\varphi), v(\psi))$;
3. $v(\varphi \Rightarrow \psi) = \sqsupset(v(\varphi), v(\psi))$;
4. $v(\neg\varphi) = -(v(\varphi))$.

2.2.2 Tautologias, contradições e contingências

Se as proposições são avaliadas através de valorações, é natural analisar as proposições sob a perspectiva das valorações. O primeiro paso é identificar duas proposições que tomam o mesmo valor em toda valoração.

Definição 2.2.7. Dizemos que duas fórmulas φ e ψ são *semanticamente equivalentes*, e o denotamos por $\varphi \equiv \psi$, se, para toda valoração v, $v(\varphi) = v(\psi)$.

A relação $\equiv$ é, de fato, uma relação de equivalência, isto é, ela é reflexiva, simétrica e transitiva (este e outros tipos de relações serão estudados no Capítulo XXX):

- $\varphi \equiv \varphi$ para toda φ (reflexiva);
- $\varphi \equiv \psi$ implica $\psi \equiv \varphi$, para toda φ, ψ (simétrica);
- $\varphi \equiv \psi$ e $\psi \equiv \gamma$ implica $\varphi \equiv \gamma$, para toda φ, ψ, γ (transitiva).

Exemplo 2.2.8. Temos que $(\varphi \vee \psi) \equiv (\psi \vee \varphi)$ e $\neg\neg\varphi \equiv \varphi$ para toda φ, ψ. $\triangle$

O mais importante subconjunto de Prop é o daquelas proposições φ que são *sempre* verdadeiras para quaisquer valorações:

Definição 2.2.9. Seja $\varphi \in \mathsf{Prop}$. Dizemos que φ é uma *tautologia* se $v(\varphi) = 1$ para toda valoração v.

As tautologias são verdades lógicas: são proposições tais que, independentemente do valor de verdade atribuido às suas componentes, recebem o valor 1 (verdadeiro). Uma maneira muito natural de determinar se uma proposição é uma tautologia é utilizando tabelas-verdade. Com efeito, a partir da Proposição 2.2.4, vemos que o valor de verdade de uma sentença φ depende exclusivamente dos valores atribuídos às variáveis que ocorrem em φ. Portanto, basta analisar todas as possíveis atribuições de valores de verdade 0 e 1 às variáveis que ocorrem em φ, combinando, para cada atribuição, os valores das variáveis (e posteriormente das sub-fórmulas) de φ de acordo com a função de verdade correspondente ao conectivo sendo utilizado.

Exemplo 2.2.10. Utilizando a representação definida acima dos conectivos de C^2 na assinatura C^0, vemos que as sentenças $p_0 \vee \neg p_0$, $(p_0 \wedge p_1) \Rightarrow p_1$ e $p_0 \Rightarrow (\neg p_0 \Rightarrow p_1)$ são tautologias:

p_0	$\neg p_0$	$p_0 \vee \neg p_0$
1	0	1
0	1	1

p_0	p_1	$p_0 \wedge p_1$	$(p_0 \wedge p_1) \Rightarrow p_1$
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	1

p_0	p_1	$\neg p_0$	$\neg p_0 \Rightarrow p_1$	$p_0 \Rightarrow (\neg p_0 \Rightarrow p_1)$
1	1	0	1	1
1	0	0	1	1
0	1	1	1	1
0	0	1	0	1

Por outro lado, as sentenças $(p_0 \vee p_1) \Rightarrow p_1$ e $((p_0 \Rightarrow p_1) \wedge p_1) \Rightarrow p_0$ não são tautologias:

p_0	p_1	$p_0 \vee p_1$	$(p_0 \vee p_1) \Rightarrow p_1$
1	1	1	1
1	0	1	0
0	1	1	1
0	0	0	1

p_0	p_1	$p_0 \Rightarrow p_1$	$(p_0 \Rightarrow p_1) \wedge p_1$	$((p_0 \Rightarrow p_1) \wedge p_1) \Rightarrow p_0$
1	1	1	1	1
1	0	0	0	1
0	1	1	1	0
0	0	1	0	1

No primeiro caso, qualquer valoração v tal que $v(p_0) = 1$ e $v(p_1) = 0$ produz $v((p_0 \vee p_1) \Rightarrow p_1) = 0$, portanto existem valorações que tornam a fórmula $\varphi = (p_0 \vee p_1) \Rightarrow p_1$ falsa, logo φ não é tautologia. Por outro lado, toda valoração v tal que $v(p_0) = 0$ e $v(p_1) = 1$ produz $v(((p_0 \Rightarrow p_1) \wedge p_1) \Rightarrow p_0) = 0$, portanto esta sentença não é uma tautologia. Esta fórmula representa uma conhecida falácia, a *falácia de afirmação do consequente*. $\triangle$

Um dos problemas mais difíceis que teremos que enfrentar é saber quando uma fórmula é uma tautologia. Além do método da tabela-verdade que acabamos de descrever (e que pode ser considerado um método sintético) há também outros métodos de tipo analítico, como o *Método de Redução ao Absurdo*. Neste método partimos da suposição de que a fórmula φ a ser testada toma o valor 0 em alguma valoração v e, utilizando as propriedades das valorações, tentamos chegar a um absurdo. Por exemplo, suponhamos que queremos determinar se a fórmula $\gamma = (\varphi \Rightarrow \psi) \Rightarrow (\neg\psi \Rightarrow \neg\varphi)$ é tautologia. Supomos, por redução ao absurdo, que $v(\gamma) = 0$ para alguma valoração v; então:

1. (a) $v(\varphi \Rightarrow \psi) = 1$, (b) $v(\neg\psi \Rightarrow \neg\varphi) = 0$
2. $v(\neg\psi) = 1$, $v(\neg\varphi) = 0$ (de 1(b))
3. $v(\psi) = 0$, $v(\varphi) = 1$ (de 2)
4. $v(\varphi \Rightarrow \psi) = 0$ (de 3)

Vemos que 4 contraria 1(a), absurdo.

Em geral, uma fórmula com n componentes atômicos necessita uma tabela com 2^n linhas para decidir se é ou não uma tautologia. Isto significa que se uma fórmula tem $n+1$ variáveis então devemos analisar $2^{n+1} = 2.2^n$ linhas. Ou seja: acrescentar apenas uma variável implica em duplicar o esforço de testar a validade da fómula!

Existiria uma maneira mais rápida de se testar tautologias?

A resposta a tal questão coincide com a solução do problema $P = NP$, um dos mais difíceis problemas da computação teórica. Pode-se demonstrar que a complexidade computacional de qualquer algoritmo se reduz a testar tautologias numa tabela-verdade. Dessa forma, se conseguíssemos provar que não existe uma maneira mais eficiente de testar todas as tautologias, ou se conseguirmos um tal método eficiente, teríamos resolvido um problema extremamente sofisticado.

O conceito dual de tautologia é o de contradição:

Definição 2.2.11. Seja $\varphi \in \mathsf{Prop}$. Dizemos que φ é uma *contradição* se $v(\varphi) = 0$ para toda valoração v.

As tautologias são falsidades lógicas: são proposições tais que, independentemente do valor de verdade atribuído às suas componentes, recebem o valor 0 (falso).

Será que tem alguma relação (ou alguma maneira de relacionar) as tautologias e as contradições? A resposta é afirmativa: a relação de dualidade é realizada através da negação $\neg$, como mostra o seguinte resultado:

Proposição 2.2.12. *(i) φ é tautologia sse $\neg\varphi$ é contradição.*
(ii) φ é contradição sse $\neg\varphi$ é tautologia.
(iii) Se φ é tautologia então existe uma contradição ψ tal que $\varphi \equiv \neg\psi$.
(iv) Se φ é contradição então existe uma tautologia ψ tal que $\varphi \equiv \neg\psi$.

Demonstração: Somente provaremos o item (iii), deixando o resto como exercício para o leitor.
(iii) Seja φ uma tautologia. Então $\psi = \neg\varphi$ é uma contradição, pelo item (i). Mas $\neg\psi = \neg\neg\varphi$ e $\neg\neg\varphi \equiv \varphi$, pelo Exemplo 2.2.8. Daqui $\varphi \equiv \neg\psi$, sendo que ψ é uma contradição. □

Vemos que existem dois tipos importantes de proposições: as sempre verdadeiras (tautologias) e as sempre falsas (contradições). Evidentemente, nem toda proposição é de um destes tipos: por exemplo, $p_1 \vee p_2$ ou p_4 são proposições que tomam ambos valores, verdadeiro e falso, em diferentes valorações: basta considerar uma valoração v tal que $v(p_1) = v(p_4) = 1$ para obter $v(p_1 \vee p_2) = v(p_4) = 1$. Por outro lado, uma valoração v' tal que $v'(p_1) = v'(p_2) = v'(p_4) = 0$ produz $v(p_1 \vee p_2) = v(p_4) = 0$. Sentenças deste tipo (que não são nem tautologias nem contradições) são chamadas de *contingências*, dado que elas são contingentes: dependendo da situação (do valor dado para as suas componentes atômicas), elas recebem ora o valor verdadeiro, ora o valor falso. Evidentemente uma proposição φ pertence a uma e só uma das três classes: tautologias, contradições, contingências.

2.2.3 Formas normais

Consideraremos nesta subseção a linguagem proposicional definida a partir da assinatura C^2. Nas fórmulas de Prop_2 podem ocorrer, portanto, os conectivos $\neg, \vee, \wedge$ e $\Rightarrow$. Veremos nesta subseção que é possível 'padronizar' as fórmulas de Prop_2, obtendo a partir delas outras fórmulas escritas numa forma padrão (chamada de forma normal), que resultam ser semanticamente equivalentes às fórmulas originais.

Antes de passar às definições, introduzimos a seguinte notação:

Notação 2.2.13. Sejam $\varphi_1, \ldots, \varphi_n$ fórmulas. Por

$$\bigvee_{i=1}^{n} \varphi_i \quad \text{e por} \quad \bigwedge_{i=1}^{n} \varphi_i$$

entendemos qualquer fórmula obtida pela disjunção (conjunção, respectivamente) das fórmulas $\varphi_1, \ldots, \varphi_n$. Isto é, não interessa a ordem e a organização dos parênteses na construção da disjunção (conjunção, respectivamente). Por convenção $\bigvee_{i=1}^{1} \varphi_i = \bigwedge_{i=1}^{1} \varphi_i = \varphi_1$.

Por exemplo, $\bigvee_{i=1}^{3} \varphi_i$ pode representar indistintamente $\varphi_1 \vee (\varphi_3 \vee \varphi_2)$ ou $(\varphi_2 \vee \varphi_1) \vee \varphi_3$.

Definição 2.2.14. (i) Um *literal* é uma fórmula da forma p (com $p \in \mathcal{V}$) ou $\neg p$ (com $p \in \mathcal{V}$).
(ii) Uma *cláusula* é uma fórmula φ da forma

$$\varphi = \bigwedge_{i=1}^{n} \psi_i$$

onde cada ψ_i é um literal e $n \geq 1$.
(iii) Dizemos que uma fórmula φ está em *forma normal disjuntiva* (FND) se φ é da forma

$$\varphi = \bigvee_{i=1}^{n} \varphi_i$$

onde cada φ_i é uma cláusula ($i = 1, \ldots, n$) e $n \geq 1$.

Por exemplo, $\varphi_1 = p_1 \wedge \neg p_1$, $\varphi_2 = (p_1 \wedge \neg p_2) \wedge \neg p_7$ e $\varphi_3 = p_6$ são cláusulas. Portanto, $\varphi_1 \vee \varphi_2 \vee \varphi_3$ é uma fórmula em forma normal.

Proposição 2.2.15. *Toda fórmula de* Prop_2 *admite uma forma normal disjuntiva. Isto é, existe uma fórmula* ψ*, nas mesmas variáveis que* φ*, tal que* ψ *está em FND, e* $\varphi \equiv \psi$*. Mais ainda, existe um algoritmo para calcular uma FND* ψ *para* φ*.*

Demonstração: Suponhamos que $q_1, \ldots, q_n$ são exatamente todas as variáveis proposicionais que ocorrem em φ. Formalmente, deve existir um número $i_j \in \mathbb{N}$ tal que $q_j = p_{i_j}$, para cada $j = 1, \ldots, n$. Construimos a tabela-verdade de φ utilizando apenas as variáveis $q_1, \ldots, q_n$. Se φ é uma contradição, definimos $\psi = \bigvee_{j=1}^{n}(q_j \wedge \neg q_j)$. Claramente ψ está em FND e é uma contradição, portanto $\varphi \equiv \psi$. Se φ não é uma contradição, dentre as 2^n linhas da tabela-verdade, escolha apenas as linhas em que φ recebe o valor 1 (deve existir ao menos uma, porque φ não é contradição). Chamemos de $L_1, \ldots, L_k$ as linhas em que φ vale 1 (note que $1 \leq k \leq 2^n$). Para cada linha L_i, defina os seguintes literais:

$$\varphi_j^i = \begin{cases} q_j & \text{se } q_j \text{ recebe o valor 1 na linha } L_i \\ \\ \neg q_j & \text{caso contrário} \end{cases}$$

Considere agora, para cada $i = 1, \ldots, k$, a cláusula $\varphi_i = \bigwedge_{j=1}^{n} \varphi_j^i$. Finalmente, definimos $\psi = \bigvee_{i=1}^{k} \varphi_i$. Resta provar que $\varphi \equiv \psi$ (pois, claramente, ψ está em FND).

Seja então v uma valoração tal que $v(\varphi) = 1$. Logo, v corresponde a uma das linhas, digamos L_i, da tabela-verdade construida para φ. Por construção dos literais φ_j^i, temos que $v(\varphi_j^i) = 1$ para todo $j = 1, \ldots, n$. Com efeito, se $v(q_j) = 1$ então $\varphi_j^i = q_j$, portanto $v(\varphi_j^i) = 1$. Por outro lado, se $v(q_j) = 0$ então $\varphi_j^i = \neg q_j$, portanto $v(\varphi_j^i) = 1$. Assim sendo, temos que $v(\varphi_i) = 1$, pois φ_i consiste da conjunção dos φ_j^i com $j = 1, \ldots, n$. Portanto, $v(\psi) = 1$, pois ψ é uma disjunção de formulas, dentre elas φ_i, sendo que $v(\varphi_i) = 1$. Reciprocamente, suponha que v uma valoração tal que $v(\psi) = 1$. Pela definição da tabela da disjunção, debe existir ao menos um índice i tal que $v(\varphi_i) = 1$, portanto $v(\varphi_j^i) = 1$ para todo $j = 1, \ldots, n$ (pela definição da tabela da conjunção). Dai inferimos que o valor dado por v para as variáveis $q_1, \ldots, q_n$ coincide com o valor dado para elas na linha L_i da tabela-verdade de φ, pela construção dos literais φ_j^i. Portanto o valor de φ (na linha L_i) coincide com $v(\varphi)$, pela Proposição 2.2.4. Logo, $v(\varphi) = 1$, pois L_i é uma das linhas em que φ recebe o valor 1. Daqui $\varphi \equiv \psi$, concluindo a demonstração. $\square$

Exemplo 2.2.16. Acharemos uma FND para $\varphi = (p_2 \vee p_3) \Rightarrow (p_1 \wedge p_3)$.

p_1	p_2	p_3	$p_2 \vee p_3$	$p_1 \wedge p_3$	$(p_2 \vee p_3) \Rightarrow (p_1 \wedge p_3)$	
1	1	1	1	1	1	L_1
1	1	0	1	0	0	
1	0	1	1	1	1	L_2
1	0	0	0	0	1	L_3
0	1	1	1	0	0	
0	1	0	1	0	0	
0	0	1	1	0	0	
0	0	0	0	0	1	L_4

Temos que as linhas relevantes são L_1 (a primeira linha), L_2 (a terceira), L_3 (a quarta) e L_4 (a oitava). Definimos, para cada uma delas, as seguintes cláusulas:

1. Para L_1 : $\varphi_1 = p_1 \wedge p_2 \wedge p_3$;
2. Para L_2 : $\varphi_2 = p_1 \wedge \neg p_2 \wedge p_3$;
3. Para L_3 : $\varphi_3 = p_1 \wedge \neg p_2 \wedge \neg p_3$;
4. Para L_4 : $\varphi_4 = \neg p_1 \wedge \neg p_2 \wedge \neg p_3$.

Finalmente, definimos $\psi = \bigvee_{i=1}^{4} \varphi_i$ como sendo uma FND para φ. $\triangle$

Existe uma outra forma normal para as fórmulas, dual da FND (no sentido de ser uma conjunção de disjunção de literais, em vez de ser uma disjunção de conjunção de literais como na FND). Esta forma normal chama-se de *forma normal conjuntiva.*

Definição 2.2.17. (i) Uma *cláusula dual* é uma fórmula φ da forma

$$\varphi = \bigvee_{i=1}^{n} \psi_i$$

onde cada ψ_i é um literal.
(iii) Dizemos que uma fórmula φ está em *forma normal conjuntiva* (FNC) se φ é da forma

$$\varphi = \bigwedge_{i=1}^{n} \varphi_i$$

onde cada φ_i é uma cláusula dual ($i = 1, \ldots, n$).

A partir da existência da FND para toda fórmula, podemos provar a existência da FNC para toda fórmula:

Proposição 2.2.18. *Toda fórmula de* Prop_2 *admite uma forma normal conjuntiva. Isto é, existe uma fórmula* ψ*, nas mesmas variáveis que* φ*, tal que* ψ *está em FNC, e* $\varphi \equiv \psi$*. Mais ainda, existe um algoritmo para calcular uma FNC* ψ *para* φ*.*

A demonstração da Proposição 2.2.18 (a partir da Proposição 2.2.15) é deixada para o leitor como exercício.

2.2.4 Conjuntos adequados de conectivos

Observe que a FND associada a φ utiliza apenas as funções $\sqcup$, $\sqcap$ e $-$. Isto significa que *qualquer* função de verdade $f : \mathbf{2}^n \to \mathbf{2}$ pode ser representada utilizando estas três funções: basta repetir o algoritmo descrito na prova da Proposição 2.2.15. Dado que as funções de verdade são representadas no nível da linguagem por conectivos, podemos dizer que esta é uma propriedade de conectivos, ou seja: existem conectivos capazes de representar qualquer função de verdade.

Definição 2.2.19. Um conjunto S de conectivos é dito um *conjunto adequado de conectivos* se as funções de verdade associadas a eles bastam para representar qualquer outra função de verdade.

Com essa definição, o conjunto $\{\neg, \vee, \wedge\}$ é adequado. Mais ainda, dado que a conjunção pode ser representada em termos de $-$ e $\sqcup$ através da equação $p \sqcap q = -(-p \sqcup -q)$, vemos que $\{\neg, \vee\}$ é um conjunto adequado de conectivos.

Proposição 2.2.20. $\{\neg, \vee\}$ *é um conjunto adequado de conectivos.*

Demonstração: Dada uma função de verdade $f(q_1, \ldots, q_n)$, considere uma FND $\psi = \bigvee_{i=1}^{n} \varphi_i$ associada a f seguindo o algoritmo da demonstração da Proposição 2.2.15. Se substituímos iteradamente em cada cláusula $\varphi_i = \bigwedge_{j=1}^{n} \varphi_j^i$ cada conjunção pela fórmula que utiliza apenas $\vee$ e $\neg$ que a representa (usando a equação $p \sqcap q = -(-p \sqcup -q)$), obteremos uma fórmula γ_i equivalente a φ_i, mas que utiliza apenas $\vee$ e $\neg$. Portanto, a disjunção $\bigvee_{i=1}^{n} \gamma_i$ é uma fórmula que representa f, e que apenas utiliza os conectivos $\neg$ e $\vee$. $\square$

É fácil provar que a disjunção pode ser representada utilizando $\wedge$ e $\neg$ através da equação $p \sqcup q = -(-p \sqcap -q)$. Assim, de maneira análoga à prova da Proposição 2.2.20, podemos provar o seguinte:

Proposição 2.2.21. $\{\neg, \wedge\}$ *é um conjunto adequado de conectivos.*

Por outro lado, da equação $p \sqcup q = -p \sqsupset q$ e da Proposição 2.2.20 obtemos imediatamente o seguinte:

Proposição 2.2.22. $\{\neg, \Rightarrow\}$ *é um conjunto adequado de conectivos.*

Consideremos agora uma constante $\mathbf{f}$ tal que $v(\mathbf{f}) = 0$ para toda valoração v. Isto é, $\mathbf{f} : \mathbf{2}^0 \to \mathbf{2}$ é dada por $\mathbf{f}(*) = 0$ (lembrando que $\mathbf{2}^0 = \{*\}$). Então $-p = p \sqsupset \mathbf{f}$ (confira!), logo obtemos o seguinte:

Proposição 2.2.23. $\{\mathbf{f}, \Rightarrow\}$ *é um conjunto adequado de conectivos.*

Por outro lado, provaremos o seguinte:

Proposição 2.2.24. $\{\neg, \Leftrightarrow\}$ *e* $\{\vee, \wedge\}$ *não são conjuntos adequados de conectivos.*

Demonstração: Seja $\varphi(p, q)$ uma função de verdade de dois argumentos p e q construída utilizando apenas as funções $-$ e $\Leftrightarrow$. Por indução na complexidade de φ (considerada como fórmula) pode ser provado que φ é tautologia, ou contradição, ou os quatro valores possíveis das quatro linhas da tabela-verdade de φ consistem exatamente de dois valores 0 e dois valores 1. Dessa maneira, vemos que não é possível definir a tabela da disjunção (os quatro valores da sua tabela consistem de três valores 1 e um valor 0) nem a da conjunção (os quatro valores da sua tabela consistem de três valores 0 e um valor 1). Portanto $\{\neg, \Leftrightarrow\}$ não é adequado.

Seja agora $\psi(p)$ uma função de um argumento p que utiliza apenas as funções $\sqcup$ e $\sqcap$ na sua composição. É fácil ver que ψ recebe o valor 1 se $p = 1$, portanto não é possível reproduzir com estas funções a negação $-$. Daqui, $\{\vee, \wedge\}$ não é adequado. $\square$

É interessante observar que os conectivos $\{\vee, \neg\}$ podem ser definidos a partir de um *único* conectivo chamado *conectivo de Sheffer* ou NAND, definido como segue: (p NAND q) $= 1$ see p e q não são simultaneamente 1:

p	q	p NAND q
1	1	0
1	0	1
0	1	1
0	0	1

Podemos definir $p \sqcup q$ e $-p$ como:

- $-p = (p \text{ NAND } p)$;
- $p \sqcup q = ((p \text{ NAND } p) \text{ NAND } (q \text{ NAND } q))$.

Desta maneira, obtemos imediatamente o seguinte:

Proposição 2.2.25. *O conjunto $\{$ NAND $\}$ é adequado.*

Similarmente, definimos um conectivo NOR dado por: $(p \text{ NOR } q) = 0$ see p e q não são simultaneamente 0:

p	q	p NOR q
1	1	0
1	0	0
0	1	0
0	0	1

Agora podemos definir $p \sqcap q$ e $-p$ como:

- $-p = (p \text{ NOR } p)$;
- $p \sqcap q = ((p \text{ NOR } p) \text{ NOR } (q \text{ NOR } q))$.

Portanto, inferimos imediatamente o seguinte:

Proposição 2.2.26. *O conjunto $\{$ NOR $\}$ é adequado.*

É possível provar o seguinte (veja [10]):

Proposição 2.2.27. *NAND e NOR são os únicos conectivos adequados.*

2.2.5 Conseqüência semântica

A partir das definições dadas até agora, podemos finalmente definir formalmente uma noção de inferência (ou de conseqüência) lógica. Dado que utilizaremos recursos semânticos para definir esta noção de conseqüência, diremos que trata-se de uma *relação de conseqüência semântica*.

Definição 2.2.28. Seja Γ um conjunto de proposições em Prop, e seja $\varphi \in \mathsf{Prop}$. Diremos que φ é *conseqüência semântica de* Γ, e o denotaremos $\Gamma \models \varphi$, se para toda valoracao v, temos o seguinte:

$$\text{se } v(\psi) = 1 \text{ para todo } \psi \in \Gamma \text{ então } v(\varphi) = 1.$$

Se $\Gamma \models \varphi$ não vale, escrevemos $\Gamma \not\models \varphi$.

Um caso particular interessante da definição anterior é quando $\Gamma = \emptyset$.

Proposição 2.2.29. $\emptyset \models \varphi$ *se e somente se* φ *é tautologia.*

Demonstração: Observe que $\emptyset \models \varphi$ sse, para toda valoração v,

$$\text{se } v(\psi) = 1 \text{ para todo } \psi \in \emptyset \text{ então } v(\varphi) = 1.$$

Assumamos que φ é conseqüência semântica de $\emptyset$. Observe que $v(\psi) = 1$ para todo $\psi \in \emptyset$ sse

$$\text{para toda } \psi, \text{ se } \psi \in \emptyset \text{ então } v(\psi) = 1.$$

Dado que é impossível exibir uma sentença ψ tal que $\psi \in \emptyset$ e $v(\psi) \neq 1$ (simplesmente porque *não tem* sentenças no conjunto vazio) então é impossível que a frase "se $\psi \in \emptyset$ então $v(\psi) = 1$" seja falsa para alguma sentença ψ. Portanto,

$$\text{para toda } \psi, \text{ se } \psi \in \emptyset \text{ então } v(\psi) = 1$$

ou, equivalentemente, "$v(\psi) = 1$ para todo $\psi \in \emptyset$" é verdadeira, para toda valoração v. Daqui, de

$$\text{"se } v(\psi) = 1 \text{ para todo } \psi \in \emptyset \text{ então } v(\varphi) = 1\text{"}$$

inferimos que $v(\varphi) = 1$, e isto vale para toda valoração v. Isto é, $\emptyset \models \varphi$ implica que $v(\varphi) = 1$ para toda valoração v. Mas isto significa que φ é uma tautologia, pela Definição 2.2.9. A recíproca é trivialmente verdadeira. $\square$

Escreveremos $\psi_1, \psi_2, \ldots, \psi_n \models \varphi$ para denotar $\{\psi_1, \psi_2, \ldots, \psi_n\} \models \varphi$. Em particular, escreveremos $\psi \models \varphi$ em lugar de $\{\psi\} \models \varphi$, e $\models \varphi$ em lugar de $\emptyset \models \varphi$. Utilizaremos também a seguinte notação: $\Gamma, \varphi \models \psi$ e $\Gamma, \Delta \models \varphi$ denotarão $\Gamma \cup \{\varphi\} \models \psi$ e $\Gamma \cup \Delta \models \varphi$, respectivamente.

Observe que podemos definir noções de conseqüência semântica $\models_1$ e $\models_2$ sobre as linguagens Prop_1 e Prop_2 utilizando as valorações definidas na Definição 2.2.6, respectivamente. Porém, o fato de $\{\vee, \neg\}$ ser adequado mostra que estas relações de conseqüência podem ser representadas por $\models$, utilizando as representações dos conectivos em função de $\{\vee, \neg\}$ que vimos anteriormente. Este fato será usado a partir de agora sem menção explícita.

Exemplos 2.2.30.

1. $\varphi, \psi \models \varphi \wedge \psi$;
2. $\varphi, \varphi \Rightarrow \psi \models \psi$;
3. $\varphi, \neg\varphi \models \psi$, para ψ qualquer;
4. $\models \varphi \Leftrightarrow \neg\neg\varphi$.

$\triangle$

As principais propriedades da relação de conseqüência semântica são as seguintes:

Teorema 2.2.31. *A relação de conseqüência $\models$ satisfaz o seguinte:*

1. $\varphi \models \varphi$ *(reflexividade);*
2. *Se* $\models \varphi$ *então* $\Gamma \models \varphi$*;*
3. *Se* $\varphi \in \Gamma$ *então* $\Gamma \models \varphi$*;*
4. *Se* $\Gamma \models \varphi$ *e* $\Gamma \subseteq \Delta$ *então* $\Delta \models \varphi$ *(monotonicidade);*
5. *Se* $\Gamma \models \varphi$ *e* $\varphi \models \psi$ *então* $\Gamma \models \psi$ *(transitividade)*
6. *Se* $\Gamma \models \varphi$ *e* $\Delta, \varphi \models \psi$ *então* $\Gamma, \Delta \models \psi$ *(corte);*
7. *Se* $\Gamma, \varphi \models \psi$ *e* $\Gamma \models \varphi$ *então* $\Gamma \models \psi$*;*
8. $\models \varphi$ *see, para todo* $\Gamma \neq \emptyset$, $\Gamma \models \varphi$*;*
9. $\Gamma, \varphi \models \psi$ *see* $\Gamma \models \varphi \Rightarrow \psi$ *(Teorema de Dedução, forma semântica);*

10. $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$ *see* $\Gamma \models \varphi_1 \Rightarrow (\varphi_2 \Rightarrow (\varphi_n \Rightarrow \varphi) \ldots)$;

11. $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$ *see* $\Gamma \models (\varphi_1 \wedge \ldots \wedge \varphi_n) \Rightarrow \varphi$.

Demonstração: Provaremos somente (8), deixando o resto da prova como exercício para o leitor:
($\rightarrow$) Segue de (2).
($\leftarrow$) Se $\Gamma \models \varphi$ para todo $\Gamma \neq \emptyset$ então, em particular, $\neg\varphi \models \varphi$. Se existir uma valoração v tal que $v(\varphi) = 0$ então $v(\neg\varphi) = 1$, portanto $v(\varphi) = 1$, pela definição de $\models$, absurdo. Portanto $v(\varphi) = 1$ para toda valoração v, isto é, $\models \varphi$. □

É muitas vezes bastante conveniente substituir as partes atômicas de alguma fórmula por outras proposições. Por exemplo, se $\models p \Rightarrow \neg\neg p$ (para $p \in \mathcal{V}$), é claro que deve valer $\models (\psi_1 \wedge \psi_2) \Rightarrow \neg\neg(\psi_1 \wedge \psi_2)$.

Essa idéia é formalizada assim: escrevemos $\varphi[\psi/p]$ para a proposição obtida trocando-se todas as ocorrências da variável p em φ por ψ.

Definição 2.2.32. Dada uma variável proposicional p e uma proposição ψ, definimos por recursão uma função *substituição de ψ em lugar de p* como a função $Subs_p^\psi : \mathsf{Prop} \rightarrow \mathsf{Prop}$ tal que:

1. $Subs_p^\psi(q) = \begin{cases} q \text{ se } q \in \mathcal{V} \text{ e } q \neq p; \\ \psi \text{ se } q \in \mathcal{V} \text{ e } q = p \end{cases}$

2. $Subs_p^\psi(\varphi_1 \vee \varphi_2) = Subs_p^\psi(\varphi_1) \vee Subs_p^\psi(\varphi_2)$;

3. $Subs_p^\psi(\neg\varphi_1) = \neg Subs_p^\psi(\varphi_1)$.

Escreveremos $\varphi[\psi/p]$ no lugar de $Subs_p^\psi(\varphi)$. A partir do Teorema 2.1.15 prova-se imediatamente que a função $Subs_p^\psi$ está bem definida, considerando $S = \mathsf{Prop}$, $H_\neg(\varphi) = \neg\varphi$, $H_\vee(\varphi_1, \varphi_2) = \varphi_1 \vee \varphi_2$, e

$$H(q) = \begin{cases} q \text{ se } q \neq p; \\ \psi \text{ se } q = p \end{cases}$$

Provaremos a seguir uma importante propriedade da relação de conseqüência semântica $\models$, a saber: o resultado de substituir componentes atômicos por partes semanticamente equivalentes produz fórmulas semanticamente equivalentes. Por exemplo, seja $\psi_1 = (p_1 \wedge p_2)$ e $\psi_2 = (\neg(p_1 \Rightarrow \neg p_2))$. Como $\psi_1 \equiv \psi_2$, então $\varphi[\psi_1/p] \equiv \varphi[\psi_2/p]$ para qualquer fórmula φ.

Teorema 2.2.33. (Teorema da Substituição, versão 1)
Sejam $\psi_1, \psi_2 \in$ Prop. Se $\psi_1 \equiv \psi_2$ então $\varphi[\psi_1/p] \equiv \varphi[\psi_2/p]$ para toda $\varphi \in$ Prop e para toda $p \in \mathcal{V}$.

Demonstração: Por indução em $l(\varphi)$.
i) Se φ é atômica: se $\varphi \neq p$, então $\varphi[\psi_1/p] = \varphi = \varphi[\psi_2/p]$ e o resultado vale trivialmente. Se $\varphi = p$ então $\varphi[\psi_1/p] = \psi_1$ e $\varphi[\psi_2/p] = \psi_2$ e o resultado vale por hipótese.
ii) Se $\varphi = \varphi_1 \vee \varphi_2$: pela hipótese de indução $\varphi_1[\psi_1/p] \equiv \varphi_1[\psi_2/p]$, e $\varphi_2[\psi_1/p] \equiv \varphi_2[\psi_2/p]$. Dado que $(\varphi_1 \vee \varphi_2)[\psi_i/p] = \varphi_1[\psi_i/p] \vee \varphi_2[\psi_i/p]$ (para $i = 1, 2$), temos que, para toda valoração v,

$$\begin{array}{rcl} v((\varphi_1 \vee \varphi_2)[\psi_1/p]) & = & v(\varphi_1[\psi_1/p] \vee \varphi_2[\psi_1/p]) \\ & = & \sqcup(v(\varphi_1[\psi_1/p]), v(\varphi_2[\psi_1/p])) \\ & = & \sqcup(v(\varphi_1[\psi_2/p]), v(\varphi_2[\psi_2/p])) \\ & = & v(\varphi_1[\psi_2/p] \vee \varphi_2[\psi_2/p]) = v((\varphi_1 \vee \varphi_2)[\psi_2/p]) \end{array}$$

portanto vale o resultado.
iii) Para o caso $\varphi = \neg\varphi_1$ o argumento é análogo. $\square$

As funções $Subs_p^\psi$ são casos particulares de um tipo de (extensão de) funções definidas por recursão, denominadas de substituições.

Definição 2.2.34. Uma *substituição* numa assinatura C é uma função $\sigma : \mathcal{V} \to L(C)$

Proposição 2.2.35. *Toda substituição σ admite uma única extensão $\widehat{\sigma} : L(C) \to L(C)$ tal que:*

1. *$\widehat{\sigma}(p) = \sigma(p)$ se $p \in \mathcal{V}$;*
2. *$\widehat{\sigma}(c(\varphi_1, \ldots, \varphi_n)) = c(\widehat{\sigma}(\varphi_1), \ldots, \widehat{\sigma}(\varphi_n))$ se $c \in C_n$ e $\varphi_1, \ldots, \varphi_n \in L(C)$.*

Demonstração: Imediata, a partir do Teorema 2.1.15. $\square$

Frequentemente escreveremos σ no lugar de $\widehat{\sigma}$, por simplicidade. Observe que $Subs_p^\psi$ nada mais é do que $\widehat{\sigma}$ para a substituição σ tal que $\sigma(p) = \psi$ e $\sigma(q) = q$ se $q \neq p$. As substituições serão utilizadas, entre outras coisas, para definir regras de inferência esquemáticas, nos próximos capítulos.

Dadas $q_1, \ldots, q_n \in \mathcal{V}$ tal que $q_i \neq q_j$ (se $i \neq j$), então as funções $Subs_p^\psi$ podem ser generalizadas para as funções $Subs_{q_1 \ldots q_n}^{\psi_1 \ldots \psi_n}$, tal que $Subs_{q_1 \ldots q_n}^{\psi_1 \ldots \psi_n}(\varphi)$

(denotado por $\varphi[\psi_1/q_1, \cdots, \psi_n/q_n]$) é a fórmula obtida de substituir *simultaneamente* a variável q_i pela fórmula ψ_i em φ. Note que, em geral, $\varphi[\psi_1/q_1, \cdots, \psi_n/q_n] \neq \varphi[\psi_1/q_1][\psi_2/q_2] \cdots [\psi_n/q_n]$ (por isso destacamos que a substituição é *simultânea*, em contraste com a substituição *iterada*). Por exemplo, $(p_1 \vee p_2)[p_2/p_1, \psi/p_2] = (p_2 \vee \psi)$ enquanto que

$$(p_1 \vee p_2)[p_2/p_1][\psi/p_2] = (p_2 \vee p_2)[\psi/p_2] = (\psi \vee \psi).$$

Note que, se σ é a substituição tal que $\sigma(q_i) = \psi_i$ (para $i = 1, \ldots, n$) e $\sigma(p) = p$ caso contrário, então $Subs^{\psi_1 \ldots \psi_n}_{q_1 \ldots q_n} = \widehat{\sigma}$. Deixamos para o leitor a prova do seguinte resultado:

Teorema 2.2.36. (Teorema da Substituição, versão 2)
Sejam $\psi_1, \ldots, \psi_n, \varphi_1, \ldots, \varphi_n \in \mathsf{Prop}$. *Se* $\psi_i \equiv \varphi_i$ *(*$i = 1, \ldots, n$*) então* $\varphi[\psi_1/q_1, \cdots, \psi_n/q_n] \equiv \varphi[\varphi_1/q_1, \cdots, \varphi_n/q_n]$ *para toda* $\varphi \in \mathsf{Prop}$ *e para todas* $q_1, \ldots, q_n \in \mathcal{V}$ *tais que* $q_i \neq q_j$ *(se* $i \neq j$*).*

2.2.6 Exercícios

1. Completar a prova da Proposição 2.2.12.

2. Provar que $\varphi \equiv \psi$ sse $\varphi \Leftrightarrow \psi$ é tautologia sse $\varphi \Rightarrow \psi$ e $\psi \Rightarrow \varphi$ são tautologias sse $\varphi \models \psi$ e $\psi \models \varphi$.

3. Usando tabelas de verdade provar que as seguintes fórmulas sao tautologias:

   (a) $(\varphi \Rightarrow \psi) \Rightarrow (\neg\psi \Rightarrow \neg\varphi)$

   (b) $(\varphi \wedge (\psi \vee \gamma)) \Leftrightarrow ((\varphi \wedge \psi) \vee (\varphi \wedge \gamma))$

4. A partir da existência da FND para toda fórmula, provar a Proposição 2.2.18 (da existência de FNC para toda fórmula).

5. Construa uma função de verdade arbitrária dependendo de três argumentos (p, q, r). Ache a FNF e a FNC dessa função de verdade.

6. Completar a prova do Teorema 2.2.31.

7. Conferir os Exemplos 2.2.30.

8. Provar que $\{\neg\varphi \Rightarrow (\psi \vee \gamma),\ \neg\psi,\ \delta \wedge \zeta,\ \gamma \Rightarrow \neg\delta\} \models \varphi$.

9. Provar a Proposição 2.2.35.

10. Provar o Teorema 2.2.36.

Capítulo 3

Axiomática e Completude

3.1 Métodos de Dedução

O método de dedução (ou prova) mais antigo que se conhece está ligado às demonstrações em Geometria, expostas nos 13 volumes dos *Elementos* de Euclides. Baseia-se na idéia de que uma prova deve começar de pontos iniciais chamados *axiomas*, e proceder por um mecanismo de síntese, gerando os teoremas através das *regras de inferência*. Esta forma de proceder foi chamada de *método axiomático* ou também *método hilbertiano*, devido ao lógico e matemático alemão David Hilbert que iniciou o estudo sistemático dos métodos de prova, sendo o precursor do que mais tarde viria a ser chamada de *teoria da prova*.

Existe um método alternativo de demonstração, baseado na *análise* das fórmulas (ao contrário do método axiomático).

Esse método, conhecido como *método dos tableaux analíticos*, tem suas raízes originariamente no estudo de Gerard Gentzen [6], discípulo de Hilbert, depois sistematizado nos trabalhos de Beth [1] e Hintikka [7]. Uma versão mais moderna é apresentada no livro de Smullyan [13], já considerado um clássico no assunto.

De certa forma, este método pode ser visto como um jogo entre dois parceiros: para demonstrar uma certa fórmula, imaginamos que o primeiro jogador tenta falsificar a fórmula. Se ele não conseguir, então a fórmula não admite contra-exemplos, isto é, não pode ser falsificada. Dessa forma o método é também conhecido como *método de prova por rejeição de todo contra-exemplo*.

O método analítico, além de extremamente elegante, é muito eficiente do ponto de vista da heurística, isto é, do ponto de vista da descoberta das provas. Isto é muito importante para a prova automática de teoremas, pois é muito mais fácil programar uma máquina para analisar fórmulas que para gerar teoremas. De um certo modo, a própria linguagem PROLOG e a programação lógica em geral fazem uso de variantes do método analítico.

Com relação a sua capacidade de provar teoremas, ambos os métodos (o método axiomático tradicional e o método analítico) são equivalentes. Isto significa que qualquer teorema que puder ser demonstrado num deles poderá sê-lo no outro. Vamos verificar que ambos têm suas vantagens e desvantagens: o método analítico é muito mais fácil de utilizar, mas é por outro lado muito mais difícil quando queremos provar *propriedades* dos métodos de prova.

Começaremos estudando o método axiomático, e no Capítulo 4 apresentaremos o método analítico em detalhes.

3.2 Sistemas Axiomáticos

Fixada uma linguagem $L(C)$, a idéia do método axiomático é definir demonstrações de fórmulas a partir de um conjunto dado de premissas. Estas demonstrações são seqüências finitas de fórmulas, cada uma delas sendo justificada pelas regras do sistema, terminando a seqüência com a fórmula que desejavamos demonstrar (caso tenhamos sucesso). As *justificativas* para colocar uma fórmula na seqüência são três: ou a fórmula é uma das premissas dadas *ad hoc*, ou trata-se de uma fórmula pertencente a um conjunto de fórmulas aceitas *a priori* pelo sistema (chamadas *axiomas*), ou ela pode ser obtida de fórmulas anteriores da seqüência por meio de uma relação entre fórmulas dada *a priori* no sistema; esta relação é uma *regra de inferência* do sistema. Formalmente:

Definição 3.2.1. Seja C uma assinatura.
(i) Uma *regra de inferência*[1] *sobre* C é um conjunto R não vazio tal que

$$R \subseteq \{\langle\{\varphi_1, \ldots, \varphi_n\}, \varphi\rangle \ : \ \{\varphi_1, \ldots, \varphi_n, \varphi\} \subseteq L(C)\}$$

para algum $n \in \mathbb{N}$. Em particular, se $\Gamma = \emptyset$ para todo $\langle\Gamma, \varphi\rangle \in R$ (isto é, se $n = 0$) então R é dita um *axioma*.
(ii) Uma regra R é *decidível* se existe um algoritmo para determinar, para

[1] Estaremos apenas considerando regras de inferência *finitárias*, isto é, com finitas premissas. Existem lógicas com regras infinitárias, mas não serão tratadas neste texto.

todo par $\langle \Gamma, \varphi \rangle$, se $\langle \Gamma, \varphi \rangle \in R$ ou $\langle \Gamma, \varphi \rangle \notin R$.[2]
(iii) Uma regra R é *esquemática* se é da forma

$$R = \{\langle \{\widehat{\sigma}(\varphi_1), \ldots, \widehat{\sigma}(\varphi_n)\}, \widehat{\sigma}(\varphi) \rangle \ : \ \sigma \text{ é uma substituição}\}$$

para algum par $\langle \{\varphi_1, \ldots, \varphi_n\}, \varphi \rangle$ fixo, chamado de *gerador de R*.

Claramente, uma regra esquemática é decidível. Observe que uma regra esquemática R pode ser (e de fato, será) substituída pelo par $\langle \{\varphi_1, \ldots, \varphi_n\}, \varphi \rangle$ que a gera. Nesse caso, um par $\langle \{\widehat{\sigma}(\varphi_1), \ldots, \widehat{\sigma}(\varphi_n)\}, \widehat{\sigma}(\varphi) \rangle$ será dito uma *instância* de R; em particular, uma fórmula $\widehat{\sigma}(\varphi)$ obtida por substituição de um axioma esquema $\langle \emptyset, \varphi \rangle$ será dita uma *instância* do axioma. Frequentemente escreveremos uma regra esquemática na forma

$$\frac{\varphi_1 \cdots \varphi_n}{\varphi} \ .$$

Em particular, um axioma esquema $\langle \emptyset, \varphi \rangle$ será escrito como φ.

Definição 3.2.2. Um *sistema axiomático* S *sobre* C é um conjunto de regras de inferência sobre C decidíveis. Se toda regra de S é esquemática diremos que S é um sistema axiomático *esquemático*. Se o conjunto S é decidível então S é dito *recursivamente axiomatizável*.

Definição 3.2.3. Seja S um sistema axiomático sobre uma assinatura C, e $\Gamma \cup \{\varphi\} \subseteq L(C)$. Uma *prova em* S *de* φ *a partir de* Γ é uma seqüência finita $\varphi_1 \cdots \varphi_n$ de fórmulas em $L(C)$ tal que $\varphi_n = \varphi$ e, para cada $1 \leq i \leq n$, vale o seguinte:

1. $\varphi_i \in \Gamma$; ou
2. existe uma regra $R \in \mathsf{S}$ e um par $\langle \Upsilon, \delta \rangle \in R$ tal que $\delta = \varphi_i$, e $\Upsilon \subseteq \{\varphi_1, \ldots, \varphi_{i-1}\}$.

Se existir uma prova em S de φ a partir de Γ escreveremos $\Gamma \vdash_{\mathsf{S}} \varphi$ e diremos que φ é uma *conseqüência sintática* de Γ. Se $\Gamma \vdash_{\mathsf{S}} \varphi$ não vale, escrevemos $\Gamma \nvdash_{\mathsf{S}} \varphi$. Se $\emptyset \vdash_{\mathsf{S}} \varphi$ diremos que φ é um *teorema* de S, e escreveremos simplesmente $\vdash_{\mathsf{S}} \varphi$. Se $\Gamma \vdash_{\mathsf{S}} \varphi$ então os elementos de Γ são chamados de *hipóteses*.

[2]Em geral, um conjunto X é *decidível* ou *recursivo* se existe um algoritmo para determinar se $x \in X$ ou $x \notin X$. Os conceitos de conjuntos, relações e funções decidíveis (ou recursivos) serão rigorosamente definidos e analisados no Capítulo XXX.

As convenções de notação da Subseção 2.2.5 para $\models$ serão adotadas também para $\vdash_{\mathsf{S}}$. Assim, $\psi_1, \psi_2, \ldots, \psi_n \vdash_{\mathsf{S}} \varphi$ (em particular, $\psi \vdash_{\mathsf{S}} \varphi$) denotará $\{\psi_1, \psi_2, \ldots, \psi_n\} \vdash_{\mathsf{S}} \varphi$; e $\Gamma, \varphi \vdash_{\mathsf{S}} \psi$ e $\Gamma, \Delta \vdash_{\mathsf{S}} \varphi$ denotarão $\Gamma \cup \{\varphi\} \vdash_{\mathsf{S}} \psi$ e $\Gamma \cup \Delta \vdash_{\mathsf{S}} \varphi$, respectivamente.

Observações 3.2.4.
(1) No caso de S ser esquemático, a cláusula 2 da Definição 3.2.3 pode ser escrita como segue:

2. existe uma regra $\langle\{\delta_1, \ldots, \delta_k\}, \delta\rangle \in \mathsf{S}$ e uma substituição σ tal que $\widehat{\sigma}(\delta) = \varphi_i$, e $\{\widehat{\sigma}(\delta_1), \ldots, \widehat{\sigma}(\delta_k)\} \subseteq \{\varphi_1, \ldots, \varphi_{i-1}\}$.

(2) Suponha que S é recursivamente axiomatizável. Embora exista um algoritmo para determinar se uma dada fórmula é ou não um axioma, e embora exista um algoritmo para determinar se é possível aplicar ou não uma regra de inferência, não existe em geral um algoritmo para decidir se uma dada fórmula é ou não um teorema. Os sistemas onde existe um tal algoritmo chamam-se de *decidíveis*. Os sistemas recursivamente axiomatizáveis, por sua vez, são ditos *semi-decidíveis*.
(3) A maioria dos sistemas axiomáticos utilizados na literatura são esquemáticos, portanto os axiomas e as regras de inferência são dados como esquemas. Por exemplo, a conhecida regra de inferência *Modus Ponens* (MP)

$$MP \quad \frac{\varphi \quad \varphi \Rightarrow \psi}{\psi}$$

nada mais é do que uma regra esquemática gerada por $\langle\{p_0, p_0 \Rightarrow p_1\}, p_1\rangle$.

A partir das definições anteriores, podemos provar o seguinte (meta)teorema sobre sistemas axiomáticos:

Teorema 3.2.5. *Se $\varphi_1, \ldots, \varphi_n \vdash_{\mathsf{S}} \varphi$ e $\vdash_{\mathsf{S}} \varphi_i$ $(i = 1, \ldots, n)$ então $\vdash_{\mathsf{S}} \varphi$.*

Demonstração: Considere uma prova em S

$$\varphi_1 \cdots \varphi_n \psi_1 \cdots \psi_m (= \varphi)$$

de φ a partir de $\{\varphi_1, \ldots, \varphi_n\}$ (note que sempre podemos colocar todas as hipóteses juntas no início de uma prova). Substituindo cada hipótese[3] φ_i por uma prova de φ_i em S, obteremos uma nova seqüência finita de fórmulas

[3]Poderia acontecer que alguma φ_i coincida com alguma ψ_j, mas não estaremos interessados em substituir essas ocorrências adicionais de φ_i.

em $L(C)$ que constitui uma prova de φ sem usar hipóteses, isto é, φ é um teorema de S. $\square$

Algumas propriedades da relação de consequência sintática são as seguintes, cuja prova deixamos como exercício para o leitor (compare com o Teorema 2.2.31).

Teorema 3.2.6. *Valem as seguintes afirmações:*
(a) $\varphi \vdash \varphi$ *(reflexividade)*
(b) Se $\vdash \varphi$ *então* $\Gamma \vdash \varphi$
(c) Se $\varphi \in \Gamma$ *então* $\Gamma \vdash \varphi$
(d) Se $\Gamma \vdash \varphi$ *e* $\Gamma \subseteq \Delta$ *então* $\Delta \vdash \varphi$ *(monotonicidade)*
(e) Se $\Gamma \vdash \varphi_i$ $(i = 1, \ldots, n)$ *e* $\varphi_1, \ldots, \varphi_n \vdash \psi$ *então* $\Gamma \vdash \psi$ *(transitividade)*
(f) Se $\Gamma \vdash \varphi$ *e* $\Delta, \varphi \vdash \psi$ *então* $\Gamma, \Delta \vdash \psi$
(g) $\Gamma \vdash \varphi$ *se e só se existe* $\Delta \subseteq \Gamma$ *(*Δ *finito) tal que* $\Delta \vdash \varphi$ *(compacidade)*
(h) Se $\Gamma, \varphi \vdash \psi$ *e* $\Gamma \vdash \varphi$ *então* $\Gamma \vdash \psi$

Uma propriedade desejável dos sistemas axiomáticos é a *estruturalidade*.

Definição 3.2.7. Um sistema axiomático S é *estrutural* se satisfaz o seguinte, para todo $\Gamma \cup \{\varphi\} \subseteq L(C)$:

$$\Gamma \vdash_{\mathsf{S}} \varphi \text{ implica que } \widehat{\sigma}(\Gamma) \vdash_{\mathsf{S}} \widehat{\sigma}(\varphi) \text{ para toda substituição } \sigma.$$

Provaremos que os sistemas axiomáticos esquemáticos são estruturais. Previamente, estabeleceremos um útil lema técnico.

Lema 3.2.8. *Sejam* σ *e* σ' *duas substituições. Considere a substituição* $\widehat{\sigma} \circ \sigma'$. *Então* $\widehat{\widehat{\sigma} \circ \sigma'} = \widehat{\sigma} \circ \widehat{\sigma'}$.

Demonstração: Por indução na complexidade da fórmula φ de $L(C)$, prova-se que

$$\widehat{\widehat{\sigma} \circ \sigma'}(\varphi) = (\widehat{\sigma} \circ \widehat{\sigma'})(\varphi).$$

Deixamos os detalhes da prova como exercício para o leitor. $\square$

Teorema 3.2.9. *Se* S *é esquemático então* S *é estrutural.*

Demonstração: Suponha que $\Gamma \vdash_{\mathsf{S}} \varphi$, e seja σ uma substituição. Por indução no comprimento n de uma prova de φ em S a partir de Γ provaremos que $\widehat{\sigma}(\Gamma) \vdash_{\mathsf{S}} \widehat{\sigma}(\varphi)$. Se $n = 1$ então temos dois casos para analisar:

Caso 1: $\varphi \in \Gamma$. Sendo assim, então $\widehat{\sigma}(\varphi) \in \widehat{\sigma}(\Gamma)$ e vale o resultado.
Caso 2: Existe uma substituição σ' e um axioma ψ em S tal que $\varphi = \widehat{\sigma'}(\psi)$. Considere a substituição $\sigma'' = \widehat{\sigma} \circ \sigma'$. Logo

$$\widehat{\sigma}(\varphi) = \widehat{\sigma}(\widehat{\sigma'}(\psi)) = (\widehat{\sigma} \circ \widehat{\sigma'})(\psi) = \widehat{\sigma''}(\psi)$$

pelo Lema 3.2.8. Daqui segue-se o resultado.

Suponhamos que o resultado vale para toda φ provada em S a partir de Γ em n passos, e suponha que φ é provada em S a partir de Γ numa prova $\varphi_1 \cdots \varphi_{n+1}$ de $n+1$ passos. Então, temos dois casos para analisar a ocorrência de $\varphi = \varphi_{n+1}$ na prova acima:
Caso 1: $\varphi \in \Gamma$ ou $\varphi = \widehat{\sigma'}(\psi)$ para algum axioma ψ de S.[4] A demonstração é como acima.
Caso 2: Existe uma regra $\langle\{\delta_1, \ldots, \delta_k\}, \delta\rangle$ em S e uma substituição σ' tal que $\widehat{\sigma'}(\delta) = \varphi$ e $\{\widehat{\sigma'}(\delta_1), \ldots, \widehat{\sigma'}(\delta_k)\} \subseteq \{\varphi_1, \ldots, \varphi_n\}$. Logo, para cada $1 \leq i \leq k$ existe $1 \leq j(i) \leq n$ tal que $\widehat{\sigma'}(\delta_i) = \varphi_{j(i)}$. Como antes, considerando a substituição $\sigma'' = \widehat{\sigma} \circ \sigma'$ temos que $\widehat{\sigma''}(\delta_i) = \widehat{\sigma}(\widehat{\sigma'}(\delta_i))$ (para $i = 1, \ldots, k$). Usando este fato e a hipótese de indução temos que, para $i = 1, \ldots, k$,

$$\widehat{\sigma}(\Gamma) \vdash_{\mathsf{S}} \widehat{\sigma''}(\delta_i),$$

pois existe uma prova de $\widehat{\sigma'}(\delta_i)$ a partir de Γ de no máximo n passos (a saber, a prova $\varphi_1 \cdots \varphi_{j(i)}$ se $\widehat{\sigma'}(\delta_i) = \varphi_{j(i)}$, com $1 \leq j(i) \leq n$). Além disso, da regra $\langle\{\delta_1, \ldots, \delta_k\}, \delta\rangle$ de S e de $\widehat{\sigma}(\varphi) = \widehat{\sigma''}(\delta)$ inferimos que

$$\widehat{\sigma''}(\delta_1), \ldots, \widehat{\sigma''}(\delta_k) \vdash_{\mathsf{S}} \widehat{\sigma}(\varphi).$$

Pelo Teorema 3.2.6(e) obtemos $\widehat{\sigma}(\Gamma) \vdash_{\mathsf{S}} \widehat{\sigma}(\varphi)$ como desejado. Isto conclui a demonstração. □

Notação 3.2.10. Seguindo a prática usual, usaremos metavariáveis para expressar as regras de inferência esquemáticas em casos concretos (como, de fato, já fizemos ao exibir o exemplo de MP acima). Assim, substituiremos as variáveis proposicionais que ocorrem na regra por metavariáveis que expressam *fórmulas arbitrárias.* Por exemplo, escreveremos

$$MP \ \ \frac{\varphi \quad \varphi \Rightarrow \psi}{\psi} \qquad \text{no lugar de} \qquad MP \ \ \frac{p_0 \quad p_0 \Rightarrow p_1}{p_1}.$$

A partir de agora, os sistemas axiomáticos que definiremos serão esquemáticos, portanto “sistema axiomático” significará “sistema axiomático esquemático”. Além disso, as regras esquemáticas de sistemas axiomáticos concretos serão formuladas utilizando metavariáveis no lugar de variáveis proposicionais.

[4] A Definição 3.2.3 de *prova* não impede escrever provas com n passos irrelevantes.

3.3 Uma axiomática para a LPC

Nesta seção definiremos um sistema axiomático para a LPC usando, como mencionado anteriormente, a linguagem Prop gerada pelos conectivos $\{\neg, \vee\}$. Daqui em diante usaremos as abreviações para os conectivos $\Rightarrow$, $\wedge$ e $\Leftrightarrow$ em função de C^0 definidas no capítulo anterior.

Definição 3.3.1. O sistema axiomático PC para a LPC consiste dos seguintes axiomas e regras de inferência:

(axioma)	$\neg\varphi \vee \varphi$
(expansão)	$\dfrac{\varphi}{\varphi \vee \psi}$
(eliminação)	$\dfrac{\varphi \vee \varphi}{\varphi}$
(associatividade)	$\dfrac{\varphi \vee (\psi \vee \gamma)}{(\varphi \vee \psi) \vee \gamma}$
(corte)	$\dfrac{\varphi \vee \psi \quad \neg\varphi \vee \gamma}{\psi \vee \gamma}$

Denotaremos as regras anteriores por (exp), (elim), (assoc) e (corte), respectivamente.

Uma longa lista de teoremas a *respeito* de teoremas é a seguinte:

Teorema 3.3.2. $\varphi, \neg\varphi \vee \psi \vdash_{PC} \psi$. *(Modus Ponens)*

Demonstração:

1. φ (hip.)
2. $\varphi \vee \psi$ (exp)
3. $\neg\varphi \vee \psi$ (hip.)
4. $\psi \vee \psi$ (corte)
5. ψ (elim) □

Teorema 3.3.3. $\varphi \vee \psi \vdash_{PC} \psi \vee \varphi$. *(Comutatividade)*

Demonstração:

1. $\varphi \vee \psi$ (hip.)
2. $\neg\varphi \vee \varphi$ (axioma)
3. $\psi \vee \varphi$ (corte) $\square$

Teorema 3.3.4. $\gamma \vee (\varphi \vee \varphi) \vdash_{PC} \gamma \vee \varphi$.

Demonstração:

1. $\gamma \vee (\varphi \vee \varphi)$ (hip.)
2. $(\gamma \vee \varphi) \vee \varphi$ (assoc)
3. $\varphi \vee (\gamma \vee \varphi)$ (3.3.3)
4. $(\varphi \vee (\gamma \vee \varphi)) \vee \gamma$ (exp)
5. $\gamma \vee (\varphi \vee (\gamma \vee \varphi))$ (3.3.3)
6. $(\gamma \vee \varphi) \vee (\gamma \vee \varphi)$ (assoc)
7. $\gamma \vee \varphi$ (elim) $\square$

Teorema 3.3.5. $(\varphi \vee \psi) \vee \gamma \vdash_{PC} \varphi \vee (\psi \vee \gamma)$.

Demonstração:

1. $(\varphi \vee \psi) \vee \gamma$ (hip.)
2. $\gamma \vee (\varphi \vee \psi)$ (3.3.3 em 1.)
3. $(\gamma \vee \varphi) \vee \psi$ ((assoc) em 2.)
4. $\psi \vee (\gamma \vee \varphi)$ (3.3.3 em 3.)
5. $(\psi \vee \gamma) \vee \varphi$ ((assoc) em 4.)
6. $\varphi \vee (\psi \vee \gamma)$ (3.3.3 em 5.) $\square$

Teorema 3.3.6. $\neg\varphi \vee \gamma, \neg\psi \vee \gamma \vdash_{PC} \neg(\varphi \vee \psi) \vee \gamma$.

Demonstração:

1. $\neg(\varphi \vee \psi) \vee (\varphi \vee \psi)$ (axioma)
2. $(\varphi \vee \psi) \vee \neg(\varphi \vee \psi)$ (3.3.3)
3. $\neg\varphi \vee \gamma$ (hip.)
4. $\varphi \vee (\psi \vee (\neg(\varphi \vee \psi))$ (3.3.5 em 2.)
5. $(\psi \vee \neg(\varphi \vee \psi)) \vee \gamma$ ((corte) em 4., 3.)
6. $\psi \vee (\neg(\varphi \vee \psi) \vee \gamma)$ (3.3.5 em 5.)
7. $\neg\psi \vee \gamma$ (hip.)
8. $(\neg(\varphi \vee \psi) \vee \gamma) \vee \gamma$ ((corte) em 6., 7.)
9. $\neg(\varphi \vee \psi) \vee (\gamma \vee \gamma)$ (3.3.5 em 8.)
10. $\neg(\varphi \vee \psi) \vee \gamma$ (3.3.4 em 9.) $\square$

Teorema 3.3.7. $\varphi \vee (\psi \vee \gamma) \vdash_{PC} \psi \vee (\gamma \vee \varphi)$.

Demonstração:
1. $\varphi \vee (\psi \vee \gamma)$ (hip.)
2. $(\psi \vee \gamma) \vee \varphi$ (3.3.3)
3. $\psi \vee (\gamma \vee \varphi)$ (3.3.5) $\square$

Teorema 3.3.8. $\varphi \vee (\psi \vee \gamma) \vdash_{PC} \gamma \vee (\varphi \vee \psi)$.

Demonstração:
1. $\varphi \vee (\psi \vee \gamma)$ (hip.)
2. $\psi \vee (\gamma \vee \varphi)$ (3.3.6)
3. $\gamma \vee (\varphi \vee \psi)$ (3.3.6) $\square$

Teorema 3.3.9. $\vdash_{PC} \neg(\varphi \vee \psi) \vee (\psi \vee \varphi)$.

Demonstração:
1. $\neg\varphi \vee \varphi$ (axioma)
2. $\varphi \vee \neg\varphi$ (3.3.3)
3. $(\varphi \vee \neg\varphi) \vee \psi$ (exp)
4. $\psi \vee (\varphi \vee \neg\varphi)$ (3.3.3)
5. $\neg\varphi \vee (\psi \vee \varphi)$ (3.3.7)
6. $\neg\psi \vee (\psi \vee \varphi)$ ((axioma), (exp), 3.3.5)
7. $\neg(\varphi \vee \psi) \vee (\psi \vee \varphi)$ (de 5. e 6., 3.3.6) $\square$

Teorema 3.3.10. $\varphi \vee (\psi \vee \gamma) \vdash_{PC} \varphi \vee (\gamma \vee \psi)$.

Demonstração:
1. $\varphi \vee (\psi \vee \gamma)$ (hip.)
2. $(\psi \vee \gamma) \vee \varphi$ (3.3.3)
3. $\neg(\psi \vee \gamma) \vee (\gamma \vee \psi)$ (3.3.9)
4. $\varphi \vee (\gamma \vee \psi)$ (de 2. e 3., (corte)) $\square$

Teorema 3.3.11. *Seja $\pi : \{\varphi, \psi, \gamma\} \to \{\varphi, \psi, \gamma\}$ uma permutação. Então:*
(a) $\varphi \vee (\psi \vee \gamma) \vdash_{PC} \pi(\varphi) \vee (\pi(\psi) \vee \pi(\gamma))$
(b) $\varphi \vee (\psi \vee \gamma) \vdash_{PC} (\pi(\varphi) \vee \pi(\psi)) \vee \pi(\gamma)$.
(c) $(\varphi \vee \psi) \vee (\varphi \vee \gamma) \vdash_{PC} \varphi \vee (\psi \vee \gamma)$.

Demonstração: Partes (a) e (b) seguem por aplicação direta de 3.3.6, 3.3.7, 3.3.8 e da regra (assoc). Para a parte (c) :

1. $(\varphi \vee \psi) \vee (\varphi \vee \gamma)$ (hip.)
2. $\varphi \vee (\psi \vee (\varphi \vee \gamma))$ (3.3.5 em 1.)
3. $\varphi \vee ((\varphi \vee \gamma) \vee \psi)$ (3.3.10 em 2.)
4. $\neg\varphi \vee (\varphi \vee \gamma)$ ((axioma), (exp) e 3.3.5)

5. $\neg\varphi \vee ((\varphi \vee \gamma) \vee \psi)$ ((exp) e 3.3.5 em 4.)
6. $(\varphi \vee \gamma) \vee \psi$ ((corte) em 3. e 5., seguido de (elim))
7. $\varphi \vee (\psi \vee \gamma)$ (3.3.5 e 3.3.10) $\square$

Teorema 3.3.12. *(Lei da dupla negação)*
(a) $\varphi \vdash_{PC} \neg\neg\varphi$.
(b) $\neg\neg\varphi \vdash_{PC} \varphi$.

Demonstração: (a) Considere a seguinte prova:
1. φ (hip.)
2. $\neg\neg\varphi \vee \neg\varphi$ (axioma)
3. $\neg\varphi \vee \neg\neg\varphi$ (3.3.3 em 2.)
4. $\neg\neg\varphi$ (3.3.2 em 1., 3.)

(b) Considere a seguinte prova:
1. $\neg\neg\varphi$ (hip.)
2. $\neg\neg\varphi \vee \varphi$ (exp)
3. $\neg\varphi \vee \varphi$ (axioma)
4. $\varphi \vee \varphi$ ((corte) em 3., 2.)
5. φ (elim) $\square$

Teorema 3.3.13. $\varphi, \neg\varphi \vdash_{PC} \psi$. *(Princípio de Pseudo-Scotus)*

Demonstração:
1. φ (hip.)
2. $\neg\varphi$ (hip.)
3. $\neg\varphi \vee \psi$ (exp)
4. ψ (3.3.2 em 1., 3.) $\square$

Provaremos a seguir o Teorema da Dedução, na forma sintática, que permite transformar a prova de uma dedução na prova de uma implicação. (Lembramos que a implicação $\varphi \Rightarrow \psi$ é uma abreviação da fórmula $\neg\varphi \vee \psi$.) É um resultado bastante útil, pois permite trabalhar com um número maior de premissas para provar fórmulas mais simples (de menor complexidade). Observe que um resultado análogo, na forma semântica, foi demonstrado no Teorema 2.2.31(9).

Teorema 3.3.14. (Teorema da Dedução, forma sintática)
$\Gamma, \varphi \vdash_{PC} \psi$ *se, e só se,* $\Gamma \vdash_{PC} \varphi \Rightarrow \psi$.

Demonstração:
($\leftarrow$) Óbvio, a partir do Teorema 3.3.2.
($\rightarrow$) Suponha que $\psi_1\psi_2\ldots\psi_n = \psi$ seja uma prova de ψ a partir de $\Gamma\cup\{\varphi\}$. Mostraremos por indução em n que $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow\psi$.

Se $n=1$ então temos dois casos para analisar:
Caso 1: Se $\psi=\widehat{\sigma}(\delta)$ para um axioma δ e uma substituição σ, ou $\psi\in\Gamma$. Então $\Gamma\vdash_{\mathsf{PC}}\psi$ e pela regra (exp) e o Teorema 3.3.3, $\Gamma\vdash_{\mathsf{PC}}\neg\varphi\vee\psi$, isto é, $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow\psi$.
Caso 2: Se $\psi=\varphi$. então $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow\varphi$, pois $\vdash_{\mathsf{PC}}\varphi\Rightarrow\varphi$ (usando o axioma $\neg\varphi\vee\varphi$).

Suponha que para toda prova $\psi_1\psi_2\ldots\psi_n$ de ψ a partir de $\Gamma\cup\{\varphi\}$ em n passos temos que $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow\psi$, e considere uma prova $\psi_1\psi_2\ldots\psi_{n+1}$ de ψ a partir de $\Gamma\cup\{\varphi\}$ em $n+1$ passos. Pela hipótese de indução, inferimos que $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow\psi_i$ para $1\leq i\leq n$. No caso em que $\psi=\widehat{\sigma}(\delta)$ para um axioma δ e uma substituição σ, ou $\psi\in\Gamma\cup\{\varphi\}$, a demonstração segue como no caso $n=1$. Se ψ é obtido dos ψ_k anteriores através das regras de derivação, temos vários casos a considerar, dependendo da regra aplicada:
(1) Se ψ foi obtido de ψ_k pela regra (exp), então $\psi=\delta\vee\gamma$, e $\psi_k=\delta$. Como $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow\delta$, i.e., $\Gamma\vdash_{\mathsf{PC}}\neg\varphi\vee\delta$ então, usando (exp) e o Teorema 3.3.5, temos que $\Gamma\vdash_{\mathsf{PC}}\neg\varphi\vee(\delta\vee\gamma)$, isto é, $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow(\delta\vee\gamma)$.
(2) Se ψ foi obtido de ψ_k pela regra (elim), então $\psi_k=\psi\vee\psi$; a prova é similar ao caso anterior, mas agora usando o Teorema 3.3.4.
(3) Se ψ foi obtido de ψ_k pela regra (assoc), então $\psi=(\delta\vee\gamma)\vee\eta$, $\psi_k=\delta\vee(\gamma\vee\eta)$. De novo usamos os teoremas 3.3.2 a 3.3.11.
(4) Se ψ foi obtido de ψ_k e ψ_j pela regra (corte), então $\psi=\gamma\vee\eta$, $\psi_k=\delta\vee\gamma$, e $\psi_j=\neg\delta\vee\eta$. Por hipótese de indução, $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow(\delta\vee\gamma)$ e $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow(\neg\delta\vee\eta)$. Aplicando os teoremas 3.3.2 a 3.3.11, obtemos: $\Gamma\vdash_{\mathsf{PC}}\delta\vee(\neg\varphi\vee\gamma)$ e $\Gamma\vdash_{\mathsf{PC}}\neg\delta\vee(\neg\varphi\vee\eta)$. Portanto segue $\Gamma\vdash_{\mathsf{PC}}(\neg\varphi\vee\gamma)\vee(\neg\varphi\vee\eta)$, aplicando (corte). Daqui, $\Gamma\vdash_{\mathsf{PC}}\neg\varphi\vee(\gamma\vee\eta)$ (usando 3.3.11(c)); isto é, $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow(\gamma\vee\eta)$.

Temos então em todos os casos $\Gamma\vdash_{\mathsf{PC}}\varphi\Rightarrow\psi$, o que completa a indução. $\square$

Corolário 3.3.15. $\Gamma\cup\{\varphi_1,\ldots,\varphi_n\}\vdash_{\mathsf{PC}}\psi$ *se, e só se,* $\Gamma\vdash_{\mathsf{PC}}\varphi_1\Rightarrow(\varphi_2\Rightarrow(\ldots\Rightarrow(\varphi_n\Rightarrow\psi)\ldots))$.

Como aplicação imediata do Teorema da Dedução, obtemos o seguinte resultado útil sobre PC:

Teorema 3.3.16.
(a) $\Gamma,\varphi\vdash_{\mathsf{PC}}\gamma$ *e* $\Gamma,\psi\vdash_{\mathsf{PC}}\gamma$ *implica* $\Gamma,\varphi\vee\psi\vdash_{\mathsf{PC}}\gamma$.
(b) $\Gamma,\varphi\vdash_{\mathsf{PC}}\gamma$ *e* $\Gamma,\neg\varphi\vdash_{\mathsf{PC}}\gamma$ *implica* $\Gamma\vdash_{\mathsf{PC}}\gamma$.

Demonstração: (a) Dado que $\Gamma, \varphi \vdash_{\mathsf{PC}} \gamma$ e $\Gamma, \psi \vdash_{\mathsf{PC}} \gamma$ então $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \gamma$ e $\Gamma \vdash_{\mathsf{PC}} \neg\psi \vee \gamma$, pelo Teorema 3.3.14. Pelo Teorema 3.3.6 temos que

$$\neg\varphi \vee \gamma, \neg\psi \vee \gamma \vdash_{\mathsf{PC}} \neg(\varphi \vee \psi) \vee \gamma,$$

portanto inferimos que $\Gamma \vdash_{\mathsf{PC}} \neg(\varphi \vee \psi) \vee \gamma$, pelo Teorema 3.2.6(e). Por outro lado, temos que

$$\varphi \vee \psi, \neg(\varphi \vee \psi) \vee \gamma \vdash_{\mathsf{PC}} \gamma$$

pelo Teorema 3.3.2. Daqui $\Gamma, \varphi \vee \psi \vdash_{\mathsf{PC}} \gamma$, pelo Teorema 3.2.6(f).
(b) Segue do item (a) e do fato de $\neg\varphi \vee \varphi$ ser um axioma de PC. □

3.4 Completude e Compacidade

Nesta seção provaremos dois teoremas importantíssimos da lógica proposicional clássica: o teorema da Completude e o teorema da Compacidade. Como veremos a seguir, ambos resultados estão interrelacionados. Dado que as técnicas para provar estes resultados são interessantes *per se*, pois elas podem ser generalizadas para provar resultados análogos para outras lógicas (chamadas de *não-clássicas*), mostraremos duas provas diferentes (ou dois caminhos alternativos para chegar à prova) do Teorema da Completude. Devemos destacar que existem duas noções de completude para um sistema lógico: a completude *fraca* e a completude *forte*. A completude fraca de um sistema S diz que $\models \varphi$ implica $\vdash_{\mathsf{S}} \varphi$, para toda fórmula φ; isto é, as tautologias (com relação a uma dada semântica) são teoremas de S. Por outro lado, a completude forte de S significa que $\Gamma \models \varphi$ implica $\Gamma \vdash_{\mathsf{S}} \varphi$, para todo conjunto de fórmulas $\Gamma \cup \{\varphi\}$. As noções de correção fraca e forte são as recíprocas das respectivas noções de completude: S é fracamente (fortemente, respectivamente) correto se $\vdash_{\mathsf{S}} \varphi$ ($\Gamma \vdash_{\mathsf{S}} \varphi$, respectivamente) implica que $\models \varphi$ ($\Gamma \models \varphi$, respectivamente). Claramente a completude forte de S implica a completude fraca de S, tomando $\Gamma = \emptyset$; o mesmo vale para a correção. No caso da completude, a recíproca não é necessariamente verdadeira (no caso que a relação de conseqüência semântica $\models$ não seja compacta); assim, um sistema pode ser fracamente completo mas não fortemente completo. Dizemos que $\models$ é *compacta* se satisfaz o seguinte: $\Gamma \models \varphi$ sse existe um subconjunto finito $\Gamma_0 \subseteq \Gamma$ de Γ tal que $\Gamma_0 \models \varphi$, para todo conjunto de fórmulas $\Gamma \cup \{\varphi\}$. Claramente, o Teorema da Completude Forte (junto com o Teorema da Correção Forte) implica o Teorema da Compacidade, pois a noção de conseqüência sintática é claramente finitária por definição (veja Teorema 3.2.6(g)). Por utro lado, a partir do Teorema da

Completude Fraca, do Teorema da Compacidade e do Teorema da Dedução obtemos facilmente o Teorema da Completude Forte.

Primeiro provaremos o Teorema da Correção.

Teorema 3.4.1. (Correção Fraca de PC) *Se* $\vdash_{PC} \varphi$ *então* $\models \varphi$.

Demonstração: Por indução no comprimento n de uma prova $\varphi_1 \cdots \varphi_n$ de φ em PC. Se $n = 1$ então φ é uma instância do axioma $\neg p_0 \vee p_0$, portanto $\models \varphi$. Suponha o resultado válido para toda φ que admite uma prova de comprimento n, e seja φ um teorema demonstrado numa seqüência $\varphi_1 \cdots \varphi_{n+1}$ de comprimento $n + 1$. Por hipótese de indução, cada φ_i $(1 \leq i \leq n)$ é uma tautologia. Claramente, dada uma instância de uma regra de PC, se as premissas da instância da regra são tautologias então a conclusão também o é (deixamos esta simples verificação como exercício para o leitor). Portanto φ_{n+1}, isto é, φ, é uma tautologia. $\square$

Corolário 3.4.2. (Correção Forte de PC) *Seja* $\Gamma \cup \{\varphi\}$ *um conjunto de fórmulas de Prop. Se* $\Gamma \vdash_{PC} \varphi$ *então* $\Gamma \models \varphi$.

Demonstração: Se $\Gamma \vdash_{PC} \varphi$ então existe um subconjunto finito $\{\varphi_1, \ldots, \varphi_n\}$ de Γ tal que $\{\varphi_1, \ldots, \varphi_n\} \vdash_{PC} \varphi$, pelo Teorema 3.2.6(g). Daqui, pelo Corolário 3.3.15, $\vdash_{PC} \varphi_1 \Rightarrow (\varphi_2 \Rightarrow (\ldots \Rightarrow (\varphi_n \Rightarrow \varphi)\ldots))$, e então $\models \varphi_1 \Rightarrow (\varphi_2 \Rightarrow (\ldots \Rightarrow (\varphi_n \Rightarrow \varphi)\ldots))$, pelo Teorema 3.4.1. Logo, pelo Teorema 2.2.31(10) inferimos que $\{\varphi_1, \ldots, \varphi_n\} \models \varphi$ e então, pelo Teorema 2.2.31(4), obtemos que $\Gamma \models \varphi$. $\square$

A primeira prova que daremos do Teorema da Completude Forte é direta, portanto tem como conseqüência imediata o Teorema da Compacidade. Esta prova é devida a Lindembaum-Asser. Para começar, introduzimos alguns conceitos e provamos alguns resultados técnicos prévios.

Definição 3.4.3. Seja $\Gamma \cup \{\varphi\}$ um conjunto de fórmulas. Dizemos que Γ é *φ-saturado* (com relação a PC) se as duas condições seguintes são verificadas:

1. $\Gamma \nvdash_{PC} \varphi$;

2. Se $\psi \notin \Gamma$ então $\Gamma, \psi \vdash_{PC} \varphi$.

Lema 3.4.4. *Seja* Γ *um conjunto* φ*-saturado. Então* Γ *satisfaz o seguinte, para toda fórmula* ψ *e* γ*:*
(a) $\psi \in \Gamma$ *sse* $\Gamma \vdash_{PC} \psi$*;*
(b) $(\psi \vee \gamma) \in \Gamma$ *sse* $\psi \in \Gamma$ *ou* $\gamma \in \Gamma$*;*

(c) $\neg\psi \in \Gamma$ *sse* $\psi \notin \Gamma$.
(d) A função característica χ_Γ *de* Γ *(isto é, a função* $\chi_\Gamma : \mathsf{Prop} \to \mathbf{2}$ *tal que* $\chi_\Gamma(\psi) = 1$ *sse* $\psi \in \Gamma$*) é uma valoração clássica (ver Definição 2.2.1).*

Demonstração: (a) Se $\psi \in \Gamma$ então evidentemente $\Gamma \vdash_{\mathsf{PC}} \psi$, pela noção de prova em sistemas axiomáticos (veja Teorema 3.2.6(c)). Reciprocamente, suponha que $\psi \notin \Gamma$. Como Γ é φ-saturado, temos que $\Gamma, \psi \vdash_{\mathsf{PC}} \varphi$. Se $\Gamma \vdash_{\mathsf{PC}} \psi$ então $\Gamma \vdash_{\mathsf{PC}} \varphi$, pelo Teorema 3.2.6(f), contradição. Portanto $\Gamma \nvdash_{\mathsf{PC}} \psi$.
(b) Suponhamos que $(\psi \vee \gamma) \in \Gamma$. Se $\psi \notin \Gamma$ e $\gamma \notin \Gamma$ então $\Gamma, \psi \vdash_{\mathsf{PC}} \varphi$ e $\Gamma, \gamma \vdash_{\mathsf{PC}} \varphi$, pois Γ é φ-saturado. Daqui obtemos que $\Gamma, \psi \vee \gamma \vdash_{\mathsf{PC}} \varphi$, pelo Teorema 3.3.16(a), e então $\Gamma \vdash_{\mathsf{PC}} \varphi$, contradição. Portanto: se $(\psi \vee \gamma) \in \Gamma$ então $\psi \in \Gamma$ ou $\gamma \in \Gamma$. Reciprocamente, suponha que $\psi \in \Gamma$. De $\psi \vdash_{\mathsf{PC}} \psi \vee \gamma$ obtemos que $\Gamma \vdash_{\mathsf{PC}} \psi \vee \gamma$, portanto $(\psi \vee \gamma) \in \Gamma$, pelo item (a). Analogamente, de $\gamma \in \Gamma$ inferirmos que $(\psi \vee \gamma) \in \Gamma$.
(c) Suponhamos que $\neg\psi \in \Gamma$. Se $\psi \in \Gamma$ então, dado que $\psi, \neg\psi \vdash_{\mathsf{PC}} \varphi$ (pelo Teorema 3.3.13), obtemos que $\Gamma \vdash_{\mathsf{PC}} \varphi$, uma contradição. Portanto: se $\neg\psi \in \Gamma$ então $\psi \notin \Gamma$. Reciprocamente, suponha que $\psi \notin \Gamma$. Então $\Gamma, \psi \vdash_{\mathsf{PC}} \varphi$. Se $\neg\psi \notin \Gamma$ então $\Gamma, \neg\psi \vdash_{\mathsf{PC}} \varphi$, portanto $\Gamma \vdash_{\mathsf{PC}} \varphi$ pelo Teorema 3.3.16(b), contradição. Logo: se $\psi \notin \Gamma$ então $\neg\psi \in \Gamma$.
(d) Seja $\chi_\Gamma : \mathsf{Prop} \to \mathbf{2}$ a função característica de Γ. Então, para toda $\psi, \gamma \in \mathsf{Prop}$ temos o seguinte: $\chi_\Gamma(\psi \vee \gamma) = 1$ sse (pela definição de χ_Γ) $(\psi \vee \gamma) \in \Gamma$ sse (pelo item (b)) $\psi \in \Gamma$ ou $\gamma \in \Gamma$ sse $\chi_\Gamma(\psi) = 1$ ou $\chi_\Gamma(\gamma) = 1$ sse $\sqcup(\chi_\Gamma(\psi), \chi_\Gamma(\gamma)) = 1$. Por outro lado, $\chi_\Gamma(\neg\psi) = 1$ sse $\neg\psi \in \Gamma$ sse (pelo item (c)) $\psi \notin \Gamma$ sse $\chi_\Gamma(\psi) = 0$ sse $-(\chi_\Gamma(\psi)) = 1$. Isto mostra que χ_Γ é uma valoração clássica sobre Prop (veja a Definição 2.2.1), concluindo a demonstração. $\square$

Lema 3.4.5. (Lindembaum-Asser) *Seja* $\Gamma \cup \{\varphi\}$ *um conjunto de fórmulas tais que* $\Gamma \nvdash_{\mathsf{PC}} \varphi$. *Então existe um conjunto* Δ *tal que* $\Gamma \subseteq \Delta$ *e* Δ *é* φ*-saturado.*

Demonstração: Considere uma enumeração $\{\varphi_n : n \in \mathbb{N}\}$ do conjunto Prop (note que Prop é enumerável, pois os conjuntos $\mathcal{V}$ e $|C^0|$ são enumeráveis). Definiremos uma seqüência $\{\Gamma_n\}_{n\in\mathbb{N}}$ de conjuntos de fórmulas da maneira seguinte: $\Gamma_0 = \Gamma$ e, para todo $n \geq 0$,

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\varphi_n\} & \text{se } \Gamma_n, \varphi_n \nvdash_{\mathsf{PC}} \varphi; \\ \\ \Gamma_n \cup \{\neg\varphi_n\} & \text{se } \Gamma_n, \varphi_n \vdash_{\mathsf{PC}} \varphi. \end{cases}$$

É fácil provar o seguinte
Fato:
(a) $\Gamma_n \subseteq \Gamma_{n+1}$ para todo $n \in \mathbb{N}$;
(b) $\Gamma_n \nvdash_{\mathsf{PC}} \varphi$ para todo $n \in \mathbb{N}$.
Com efeito, (a) é imediato da definição dos conjuntos Γ_n. Provaremos (b) por indução em n. Se $n = 0$ então o resultado vale pela hipótese sobre Γ. Suponha que $\Gamma_n \nvdash_{\mathsf{PC}} \varphi$, para um dado $n \geq 0$. Se $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$ então claramente $\Gamma_{n+1} \nvdash_{\mathsf{PC}} \varphi$, pela definição de Γ_{n+1}. Por outro lado, se $\Gamma_{n+1} = \Gamma_n \cup \{\neg\varphi_n\}$ então $\Gamma_n, \varphi_n \vdash_{\mathsf{PC}} \varphi$, pela definição de Γ_{n+1}. Suponha que $\Gamma_{n+1} \vdash_{\mathsf{PC}} \varphi$. Então, pelo Teorema 3.3.16(b), obtemos que $\Gamma_n \vdash_{\mathsf{PC}} \varphi$, contradição. Logo $\Gamma_{n+1} \nvdash_{\mathsf{PC}} \varphi$. Isto prova o **Fato**.
Considere agora o conjunto $\Delta = \bigcup_{n\in\mathbb{N}} \Gamma_n$. Claramente $\Gamma \subseteq \Delta$. Provaremos a seguir que Δ é φ-saturado. Suponha que $\Delta \vdash_{\mathsf{PC}} \varphi$. Pelo Teorema 3.2.6(g) temos que $\Delta_0 \vdash_{\mathsf{PC}} \varphi$ para algum subconjunto finito Δ_0 de Δ. Pela definição de Δ e pelo **Fato**(a) inferimos que $\Gamma_n \vdash_{\mathsf{PC}} \varphi$ para algum n suficientemente grande. Mas isto contradiz o **Fato**(b). Portanto $\Delta \nvdash_{\mathsf{PC}} \varphi$. Seja agora ψ uma fórmula em Prop tal que $\psi \notin \Delta$. Existe um número m tal que $\psi = \varphi_m$, pois partimos de uma enumeração de Prop. Dado que $\varphi_m \notin \Delta$ então $\varphi_m \notin \Gamma_{m+1}$, portanto $\Gamma_{m+1} = \Gamma_m \cup \{\neg\varphi_m\}$ e $\Gamma_m, \varphi_m \vdash_{\mathsf{PC}} \varphi$, pela construção da seqüência de conjuntos Γ_n. Mas então obtemos que $\Delta, \varphi_m \vdash_{\mathsf{PC}} \varphi$, isto é, $\Delta, \psi \vdash_{\mathsf{PC}} \varphi$. Isto prova que Δ é φ-saturado, concluindo a demonstração. □

Teorema 3.4.6. (Completude Forte de PC) *Seja $\Gamma \cup \{\varphi\}$ um conjunto de fórmulas em Prop. Então $\Gamma \models \varphi$ implica $\Gamma \vdash_{\mathsf{PC}} \varphi$.*

Demonstração: Suponhamos que $\Gamma \nvdash_{\mathsf{PC}} \varphi$. Então existe um conjunto φ-saturado Δ tal que $\Gamma \subseteq \Delta$, pelo Lema 3.4.5. Seja $\chi_\Delta : \mathsf{Prop} \to \mathbf{2}$ a função característica do conjunto Δ; logo, χ_Δ é uma valoração, pelo Lema 3.4.4(d). Mais ainda, $\chi_\Delta(\psi) = 1$ para toda $\psi \in \Gamma$ (pois $\Gamma \subseteq \Delta$), e $\chi_\Delta(\varphi) = 0$, pois $\varphi \notin \Delta$ (pelo Teorema 3.2.6(c)). Logo, $\Gamma \not\models \varphi$, pela definição de conseqüência semântica 2.2.28. Ou seja: se $\Gamma \models \varphi$ então $\Gamma \vdash_{\mathsf{PC}} \varphi$, como queríamos demonstrar. □

Corolário 3.4.7. (Adequação Forte de PC) *O sistema PC é fortemente adequado, isto é, PC é fortemente correto e fortemente completo com relação à semântica de valorações: para todo conjunto de fórmulas $\Gamma \cup \{\varphi\}$ em Prop temos que $\Gamma \models \varphi$ sse $\Gamma \vdash_{\mathsf{PC}} \varphi$.*

Como corolário do Teorema da Adequação Forte, obtemos o Teorema da Compacidade, que afirma que a relação de conseqüência semântica é finitária, no sentido em que uma sentença é semanticamente dedutível de no máximo um conjunto finito de sentenças ou, em outras palavras, conjuntos infinitos de premissas são desnecessários para a relação de conseqüência semântica (observe que uma versão sintática deste resultado aparece no Teorema 3.2.6(g)).

Corolário 3.4.8. (Teorema da Compacidade da LPC) *Seja* $\Gamma \cup \{\varphi\}$ *um conjunto de fórmulas em* Prop*. Se* $\Gamma \models \varphi$ *então existe um subconjunto finito* Γ_0 *de* Γ *tal que* $\Gamma_0 \models \varphi$*.*

Demonstração: Se $\Gamma \models \varphi$ então $\Gamma \vdash_{\mathsf{PC}} \varphi$, pelo Teorema 3.4.6. Daqui, existe um subconjunto finito Γ_0 de Γ tal que $\Gamma_0 \vdash_{\mathsf{PC}} \varphi$, pelo Teorema 3.2.6(g). Logo, $\Gamma_0 \models \varphi$ pelo Corolário 3.4.2. □

É possível dar uma outra formulação equivalente do teorema da compacidade, que às vezes resulta útil. Antes de dar uma formulação equivalente da compacidade semântica, precisamos de umas definições e resultados prévios.

Definição 3.4.9. Dado um conjunto de fórmulas $\Gamma \subseteq \mathsf{Prop}$, dizemos que Γ é *satisfatível* se existe uma valoração $v : \mathsf{Prop} \to \mathbf{2}$ tal que $v(\psi) = 1$ para toda $\psi \in \Gamma$. Caso contrário, isto é, se para toda valoração v existe uma fórmula $\psi \in \Gamma$ tal que $v(\psi) = 0$, então dizemos que Γ é *insatisfatível.*

Proposição 3.4.10. *Seja* $\Gamma \cup \{\varphi\}$ *um conjunto de fórmulas em* Prop*. Então* $\Gamma \models \varphi$ *sse* $\Gamma \cup \{\neg\varphi\}$ *é insatisfatível.*

Demonstração: Suponha que $\Gamma \cup \{\neg\varphi\}$ é satisfatível. Logo, existe uma valoração v tal que $v(\psi) = 1$ para toda $\psi \in \Gamma$, e $v(\neg\varphi) = 1$. Logo, a valoração v é tal que $v(\psi) = 1$ para toda $\psi \in \Gamma$, e $v(\varphi) = 0$, portanto $\Gamma \not\models \varphi$. Ou seja: se $\Gamma \models \varphi$ então $\Gamma \cup \{\neg\varphi\}$ é insatisfatível. Reciprocamente, suponha que $\Gamma \not\models \varphi$. Logo, existe alguma valoração v tal que $v(\psi) = 1$ para toda $\psi \in \Gamma$, e $v(\varphi) = 0$ (portanto $v(\neg\varphi) = 1$). Logo, o conjunto $\Gamma \cup \{\neg\varphi\}$ é satisfatível. Daqui: se $\Gamma \cup \{\neg\varphi\}$ é insatisfatível então $\Gamma \models \varphi$. □

Proposição 3.4.11. *São equivalentes:*
(a) Para todo subconjunto $\Gamma \cup \{\varphi\}$ *de* Prop*: se* $\Gamma \models \varphi$ *então existe um subconjunto finito* Γ_0 *de* Γ *tal que* $\Gamma_0 \models \varphi$*.*
(b) Para todo subconjunto Γ *de* Prop*: se todo subconjunto finito de* Γ *é satisfatível, então* Γ *é satisfatível.*

Demonstração: $(a) \to (b)$: Observe que (b) equivale à seguinte afirmação:

$(b)'$ Para todo subconjunto Γ de Prop: se Γ é insatisfatível então existe um subconjunto finito Γ_0 de Γ tal que Γ_0 é insatisfatível.

Provaremos que (a) implica $(b)'$. Seja então Γ um conjunto insatisfatível; logo, $\Gamma \models \neg(p_0 \vee \neg p_0)$ (isto sai claramente da definição de conseqüência semântica). Usando (a) temos que $\Gamma_0 \models \neg(p_0 \vee \neg p_0)$ para algum subconjunto finito Γ_0 de Γ, portanto $\Gamma_0 \cup \{\neg\neg(p_0 \vee \neg p_0)\}$ é insatisfatível, pela Proposição 3.4.10. Mas, dado que $\neg\neg(p_0 \vee \neg p_0)$ é uma tautologia, então certamente Γ_0 é insatisfatível, sendo que Γ_0 é um subconjunto finito de Γ. Isto prova $(b)'$, portanto (b).
$(b) \to (a)$: Observe que, usando a Proposição 3.4.10, temos que (a) equivale ao seguinte:

$(a)'$ Para todo subconjunto $\Gamma \cup \{\varphi\}$ de Prop: se $\Gamma \cup \{\neg\varphi\}$ é insatisfatível, então existe um subconjunto finito Γ_0 de Γ tal que $\Gamma_0 \cup \{\neg\varphi\}$ é insatisfatível.

Podemos ainda reescrever $(a)'$ da seguinte maneira equivalente:

$(a)''$ Para todo subconjunto $\Gamma \cup \{\varphi\}$ de Prop: se para todo subconjunto finito Γ_0 de Γ temos que $\Gamma_0 \cup \{\neg\varphi\}$ é satisfatível, então $\Gamma \cup \{\neg\varphi\}$ é satisfatível.

Provaremos agora que (b) implica $(a)''$. Suponha então que para todo subconjunto finito Γ_0 de Γ temos que $\Gamma_0 \cup \{\neg\varphi\}$ é satisfatível, e seja $\Delta = \Gamma \cup \{\neg\varphi\}$. Seja $\Delta_0 \subseteq \Delta$ finito, e considere $\Gamma_0 = \Delta_0 - \{\neg\varphi\}$ (note que possivelmente $\neg\varphi \notin \Delta_0$). Então Γ_0 é um subconjunto finito de Γ, portanto $\Gamma_0 \cup \{\neg\varphi\}$ é satisfatível, por hipótese. Dado que $\Delta_0 \subseteq \Gamma_0 \cup \{\neg\varphi\}$ (a igualdade vale sse $\neg\varphi \in \Delta_0$) então inferimos que Δ_0 é satisfatível, e isto vale para todo subconjunto finito Δ_0 de Δ. Usando (b) inferimos que Δ é satisfatível. Isto prova $(a)''$, portanto (a). □

Daremos agora uma segunda prova *construtiva* de completude (fraca) do sistema PC (a primeira, não explicitada aqui, é um caso particular da prova da completude forte, para $\Gamma = \emptyset$), baseando-se numa prova proposta por Kálmar em 1935.

Lema 3.4.12. *Seja φ uma fórmula, e $q_1, \ldots, q_n$ as variáveis proposicionais que ocorrem em φ. Seja v uma valoração qualquer. Para $1 \leq i \leq n$ definimos*

$$q_i^\star = \begin{cases} q_i & \textit{se} \;\; v(q_i) = 1 \\ \neg q_i & \textit{se} \;\; v(q_i) = 0 \end{cases}$$

e seja $\Delta_\varphi = \{q_1^\star, q_2^\star, \ldots, q_n^\star\}$*. Então:*

(i) Se $v(\varphi) = 1$ *então* $\Delta_\varphi \vdash_{\mathsf{PC}} \varphi$*, e*
(ii) Se $v(\varphi) = 0$ *então* $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\varphi$*.*

Demonstração: Por indução na complexidade n de φ. Se $n = 1$ então φ é q_1, portanto $\Delta_\varphi = \{q_1\}$ ou $\Delta_\varphi = \{\neg q_1\}$, e o problema se reduz a mostrar que $q_1 \vdash_{\mathsf{PC}} q_1$ ou $\neg q_1 \vdash_{\mathsf{PC}} \neg q_1$, o que é trivialmente verdadeiro. Suponha o resultado válido para toda φ com $l(\varphi) \leq n$ para certo $n \geq 1$, e seja φ tal que $l(\varphi) = n + 1$. Temos dois casos:
(1) Se φ é $\neg\psi$. Note que $\Delta_\varphi = \Delta_\psi$. Suponha que $v(\varphi) = 1$. Então $v(\psi) = 0$ e por hipótese de indução $\Delta_\psi \vdash_{\mathsf{PC}} \neg\psi$, isto é, $\Delta_\varphi \vdash_{\mathsf{PC}} \varphi$. No caso de ter $v(\varphi) = 0$ então $v(\psi) = 1$ e pela hipótese de indução $\Delta_\psi \vdash_{\mathsf{PC}} \psi$. Como $\psi \vdash_{\mathsf{PC}} \neg\neg\psi$ concluimos que $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\neg\psi$, pelo Teorema 3.2.6(e).
(2) Se φ é $\psi \vee \gamma$. Então $\Delta_\psi \subseteq \Delta_\varphi$ e $\Delta_\gamma \subseteq \Delta_\varphi$. Suponhamos que $v(\varphi) = 1$. Então $v(\psi) = 1$ ou $v(\gamma) = 1$. No primeiro caso, pela hipótese de indução, $\Delta_\psi \vdash_{\mathsf{PC}} \psi$, e daí $\Delta_\varphi \vdash_{\mathsf{PC}} \psi \vee \gamma$ pela regra (exp) e pelo Teorema 3.2.6(d). O segundo caso é análogo. Suponhamos agora que $v(\varphi) = 0$. Então $v(\psi) = v(\gamma) = 0$, e pela hipótese de indução, $\Delta_\psi \vdash_{\mathsf{PC}} \neg\psi$ e $\Delta_\gamma \vdash_{\mathsf{PC}} \neg\gamma$. Em conseqüência, pela regra (exp) e pelo Teorema 3.2.6(d) temos:

$$\Delta_\varphi \vdash_{\mathsf{PC}} \neg\psi \vee \neg(\psi \vee \gamma) \ \text{ e } \ \Delta_\varphi \vdash_{\mathsf{PC}} \neg\gamma \vee \neg(\psi \vee \gamma).$$

Pelo Teorema 3.3.6, $\Delta_\varphi \vdash_{\mathsf{PC}} \neg(\psi \vee \gamma) \vee \neg(\psi \vee \gamma)$ e então $\Delta_\varphi \vdash_{\mathsf{PC}} \neg(\psi \vee \gamma)$, pela regra (elim). □

Teorema 3.4.13. (Completude Fraca de PC) *Seja* φ *uma fórmula em Prop. Logo, temos que: se* $\models \varphi$ *então* $\vdash_{\mathsf{PC}} \varphi$*.*

Demonstração: Seja φ uma tautologia; então $v(\varphi) = 1$ para toda valoração v. Seja $Var(\varphi) = \{q_1, \ldots, q_n\}$ o conjunto das variáveis proposicionais que ocorrem em φ. Considere duas valorações v_1 e v_2 tais que v_1 atribui 1 a todas as variáveis proposicionais de φ e v_2 atribui 1 a todas, exceto a q_n, isto é, $v_2(q_n) = 0$. Pelo lema anterior, temos:

$$\{q_1, q_2, \ldots, q_{n-1}, q_n\} \vdash_{\mathsf{PC}} \varphi \ \text{ e } \ \{q_1, q_2, \ldots, q_{n-1}, \neg q_n\} \vdash_{\mathsf{PC}} \varphi.$$

Pelo Teorema 3.3.16(b) obtemos

$$(*) \ \ \{q_1, q_2, \ldots, q_{n-1}\} \vdash_{\mathsf{PC}} \varphi.$$

Repetindo este processo para valorações v_3 e v_4 tais que $v_3(q_i) = 1$ $(1 \leq i \leq n-2)$, $v_3(q_{n-1}) = 0$, $v_3(q_n) = 1$, e $v_4(q_i) = 1$ $(1 \leq i \leq n-2)$, $v_4(q_{n-1}) = 0$, $v_4(q_n) = 0$, temos, da mesma forma,

$$(**) \quad \{q_1, q_2, \ldots, \neg q_{n-1}\} \vdash_{\mathsf{PC}} \varphi.$$

Consequentemente, aplicando Teorema 3.3.16(b) em $(*)$ e $(**)$, obtemos

$$(***) \quad \{q_1, q_2, \ldots, q_{n-2}\} \vdash_{\mathsf{PC}} \varphi.$$

Observe que, utilizando valorações v_1 e v_2 conseguimos eliminar q_n em $(*)$, e em seguida utilizando v_3 e v_4 pudemos eliminar também q_{n-1} em $(***)$. Fica claro, entao, que repetindo este procedimento com 2^n valorações, obteremos $\vdash_{\mathsf{PC}} \varphi$. $\square$

Observe que, neste ponto, para obter uma outra prova do Teorema de Completude Forte (a partir do Teorema de Completude Fraca) deveriamos primeiro dar uma prova independente do Teorema da Compacidade (isto é, que não utilize o Teorema de Completude Forte, como nós fizemos na prova do Corolário 3.4.8). Usando estes resultados (Completude Fraca e Compacidade semântica) junto com o Teorema da Dedução (sintático e semântico) conseguiriamos então uma outra prova do Teorema de Completude Forte. Deixamos os detalhes como exercício para o leitor.

3.5 Outras Axiomáticas

A axiomática PC que utilizamos na seção anterior possui apenas um axioma e diversas regras de inferência. Existem muitas outras axiomáticas, considerando a mesma linguagem (ou otras linguagens apropriadas) da LPC, mas com diferentes axiomas, e em geral utilizando apenas a regra de *Modus Ponens*. O leitor interessado pode encontrar diversas axiomáticas para a lógica proposicional no livro [5].

A seguinte axiomática foi popularizada no conhecido livro de Mendelson [10], embora pertença a Hilbert e Bernays, e é sem dúvidas uma das axiomáticas da LPC mais divulgadas na literatura:

Definição 3.5.1. O sistema axiomático M para a LPC escrito na linguagem Prop_1 consiste dos seguintes axiomas e regras de inferência:

(Axioma 1) $\varphi \Rightarrow (\psi \Rightarrow \varphi)$

(Axioma 2) $(\varphi \Rightarrow (\psi \Rightarrow \gamma)) \Rightarrow ((\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \gamma))$

(Axioma 3) $(\neg\psi \Rightarrow \neg\varphi) \Rightarrow ((\neg\psi \Rightarrow \varphi) \Rightarrow \psi)$

(Modus Ponens) $\dfrac{\varphi \quad \varphi \Rightarrow \psi}{\psi}$

É possível mostrar que o sistema M equivale ao sistema PC, no sentido em que cada regra de M pode ser demonstrada em PC, e vice-versa, com as respectivas traduções entre linguagens: $\varphi \Rightarrow \psi$ é representado como $\neg\varphi \vee \psi$ em Prop, e $\varphi \vee \psi$ é representado como $\neg\varphi \Rightarrow \psi$ em Prop_1. Portanto, M satisfaz o Teorema da Dedução e o Teorema da Adequação Forte com relação à semântica de valorações da Definição 2.2.6(i).

Um outro tipo de axiomática que pode-se introduzir lida diretamente com conjuntos de fórmulas (cf. [3]). Podemos interpretar intuitivamente um conjunto Γ de fórmulas como a disjunção de seus elementos. Dessa forma, para Γ finito, "Γ, φ" denota a disjunção dos elementos de $\Gamma \cup \{\varphi\}$. Lembremos que $\varphi \wedge \psi$ é representado por $\neg(\neg\varphi \vee \neg\psi)$ em Prop.

Definição 3.5.2. O sistema CR definido sobre C^0 é da seguinte forma:

Axiomas: $\Gamma, \varphi, \neg\varphi$

Regras:

$$R_1 \quad \frac{\Gamma, \varphi}{\Gamma, \varphi \vee \psi} \qquad\qquad R_2 \quad \frac{\Gamma, \psi}{\Gamma, \varphi \vee \psi}$$

$$R_3 \quad \frac{\Gamma, \varphi \qquad \Gamma, \psi}{\Gamma, \varphi \wedge \psi} \qquad\qquad R_4 \quad \frac{\Gamma, \varphi}{\Gamma, \neg\neg\varphi}$$

$$\mathit{Corte} \quad \frac{\Gamma, \varphi \qquad \Gamma, \neg\varphi}{\Gamma}$$

Pode-se também demonstrar que o sistema CR é equivalente ao sistema PC (e a todos os outros) no sentido seguinte: $\vdash_{\mathsf{CR}} \{\varphi_1, \ldots, \varphi_n\}$ se e somente se $\vdash_{\mathsf{PC}} \bigvee_{i=1}^{n} \varphi_i$.

O sistema CR é muito interessante pela seguinte razão: nele a Regra do Corte pode ser eliminada. A primeira situação desse tipo foi demonstrada por G. Gentzen, num teorema conhecido como *Hauptsatz* (veja o Teorema 4.3.4), que teve conseqüências profundas para os fundamentos da matemática.

Finalizamos esta seção mencionando um teorema importante (e surpreendente) da lógica proposicional, o *Teorema da Interpolação de Craig*:

Teorema 3.5.3. (Teorema da Interpolação de Craig) *Sejam $\varphi_1, \varphi_2 \in$ Prop tais que φ_1 não é contradição e φ_2 não é tautologia. Se $\models \varphi_1 \Rightarrow \varphi_2$, existe $\varphi \in$ Prop tal que $Var(\varphi) \subseteq Var(\varphi_1) \cap Var(\varphi_2)$, $\models \varphi_1 \Rightarrow \varphi$ e $\models \varphi \Rightarrow \varphi_2$.*

(A prova de este teorema é deixada como exercício para o leitor.) O que este teorema afirma é que entre cada duas fórmulas separadas por implicação tais que a implicação é um teorema, sempre se pode interpolar outra fórmula φ de modo a se obter dois novos teoremas. É interessante observar que uma propriedade semelhante ocorre com os números racionais: entre cada dois racionais pode-se colocar mais um; esta propriedade é chamada *densidade*. De certa forma, podemos pensar que os teoremas da lógica proposicional formam um conjunto denso.

3.6 Axiomáticas não completas

Finalizamos este capítulo mostrando duas axiomatizações não completas da LPC. Estes dois exemplos servirão para mostrar duas coisas: (1) os cuidados que devemos tomar ao tentar axiomatizar uma determinada lógica (no caso, a LPC); e (2) uma técnica que pode ser utilizada para provar que um sistema é incompleto ou, mas geralmente, para provar que uma determinada fórmula *não é demonstrável* num dado sistema axiomático. Portanto, esta técnica serve também para provar a independência dos axiomas de uma dada axiomática, isto é: para provar que o sistema obtido de eliminar um dos seus axiomas não consegue provar o axioma eliminado.

Nesta seção , "Completude" e "Correção" referem-se a "Completude Fraca" e "Correção Fraca", respectivamente.

Definição 3.6.1. Considere a assinatura proposicional C^4 dada por $C_0^4 = \{\mathbf{f}\}$, $C_2^4 = \{\vee, \Rightarrow\}$ e $C_n^4 = \emptyset$ nos outros casos. O sistema axiomático P escrito na linguagem $L(C^4)$ consiste dos seguintes axiomas e regras de inferência:

(Ax1) $(\varphi \vee \varphi) \Rightarrow \varphi$

(Ax2) $\varphi \Rightarrow (\psi \vee \varphi)$

(Ax3) $(\varphi \Rightarrow \psi) \Rightarrow ((\gamma \vee \varphi) \Rightarrow (\psi \vee \gamma))$

(MP) $\dfrac{\varphi \qquad \varphi \Rightarrow \psi}{\psi}$

Definição 3.6.2. Uma *valoração para* P é uma função $v : L(C^4) \to \mathbf{2}$ tal que:

1. $v(\mathbf{f}) = 0$;
2. $v(\varphi \vee \psi) = \sqcup(v(\varphi), v(\psi))$;
3. $v(\varphi \Rightarrow \psi) = \sqsupset(v(\varphi), v(\psi))$,

onde $\sqcup : \mathbf{2}^2 \to \mathbf{2}$ e $\sqsupset$: $\mathbf{2}^2 \to \mathbf{2}$ são as funções de verdade clássicas (veja Subseção 2.2.1).

Observe que a semântica de P coincide com a semântica clássica para os conectivos $\{\mathbf{f}, \vee, \Rightarrow\}$ (um conjunto adequado), portanto as tautologias de P (nesta linguagem e nas linguagens Prop, Prop_1 e Prop_2, se usarmos as abreviações usuais) são as clássicas.

Proposição 3.6.3. *O sistema P é correto com relação às suas valorações.*

Demonstração: Imediata: os axiomas são tautologias clássicas, e a regra (MP) preserva validade: se as premissas são verdadeiras, então a conclusão é também verdadeira. □

Porém, podemos observar que o sistema P não é completo com relação à sua semântica clássica:

Proposição 3.6.4. *O sistema P não é completo com relação às suas valorações.*

Demonstração: Basta provar que existe uma tautologia (com relação às suas valorações, ou seja, uma tautologia clássica na linguagem $L(C^4)$) que não é demonstrável em P. Considere a fórmula $\varphi = (\mathbf{f} \Rightarrow p_0)$. Claramente

φ é uma tautologia. Provaremos que φ não é demonstrável em P usando a técnica seguinte: exibiremos uma semântica alternativa para P que valide seus axiomas e suas regras de inferência (portanto, P será correto para essa semântica alternativa). Porém, φ não será uma tautologia com relação a essa semântica, portanto não poderá ser um teorema de P. Considere então a seguinte semântica para P: as valorações são funções como na Definição 3.6.2, mas agora $v(\mathbf{f}) = 1$ para toda v. Dado que $\mathbf{f}$ não aparece explicitamente nos axiomas e nas regras de P, vemos que P é correto com relação a esta nova semântica. Mas, claramente, φ não é tautologia com relação a esta semântica, pois para qualquer valoração v tal que $v(p_0) = 0$ temos que $v(\varphi) = 0$. Portanto φ não pode ser um teorema de P, pela correção de P com relação a esta semântica. $\square$

Um motivo óbvio na falha de P para axiomatizar a LPC é a ausência de axiomas e regras governando o símbolo primitivo $\mathbf{f}$ da linguagem de P. Um fato intuitivo básico com relação à axiomatização de sistemas lógicos é que deveriamos colocar regras e axiomas para cada conectivo declarado na assinatura, caso contrário nos vemos expostos a fenómenos de incompletude como o que acabamos de apresentar.

Mas as coisas não são tão fáceis, e mesmo colocando axiomas que representam as inferências básicas de uma lógica intuitivamente suficientes para gerá-la, podemos ter surpressas. Mostraremos a seguir uma extensão de P acrescentando alguns axiomas para a constante $\mathbf{f}$ e para a negação derivada $\neg\varphi := (\varphi \Rightarrow \mathbf{f})$, que ainda resultará ser incompleta. A prova de incompletude é análoga à prova da Proposição 3.6.4, mas agora utilizaremos uma semântica trivalente, isto é, com três valores de verdade.

Definição 3.6.5. O sistema P^+ definido sobre $L(C^4)$ é a extensão de P obtida pelo acréscimo dos seguintes axiomas a P (onde $\neg\varphi$ denota $\varphi \Rightarrow \mathbf{f}$):

(Ax4) $\mathbf{f} \Rightarrow \varphi$

(Ax5) $\varphi \Rightarrow (\neg\varphi \Rightarrow \psi)$

(Ax6) $\neg\neg\varphi \Rightarrow \varphi$

(Ax7) $\varphi \Rightarrow \neg\neg\varphi$

(Ax8) $\neg\varphi \vee \varphi$

Primeiro de tudo, observemos que os acréscimos feitos ao sistema P são classicamente corretos.

Proposição 3.6.6. *O sistema P^+ é correto com relação à semântica clássica de valorações v tais que $v(\mathbf{f}) = 0$.*

Demonstração: Imediata. $\square$

Para provar a incompletude de P^+ com relação à semântica clássica introduzimos uma semântica alternativa para o sistema, tal que P^+ é correto com relação a essa semântica; porém, essa semântica não valida uma certa tautologia clássica.

Definição 3.6.7. Seja $\mathbf{3} = \{0, \frac{1}{2}, 1\}$. Uma *valoração* para P^+* é uma função $v : L(C^4) \to \mathbf{3}$ tal que:

1. $v(\mathbf{f}) = 0$;
2. $v(\varphi \vee \psi) = \sqcup(v(\varphi), v(\psi))$;
3. $v(\varphi \Rightarrow \psi) = \sqsupset(v(\varphi), v(\psi))$,

onde $\sqcup : \mathbf{3}^2 \to \mathbf{3}$ e $\sqsupset$: $\mathbf{3}^2 \to \mathbf{3}$ são as funções de verdade definidas pelas matrices abaixo.

$\sqcup$	1	$^1/_2$	0
1	1	1	1
$^1/_2$	1	0	0
0	1	0	0

$\sqsupset$	1	$^1/_2$	0
1	1	0	0
$^1/_2$	1	$^1/_2$	1
0	1	1	1

(Aqui, o primeiro argumento das funções $\sqcup$ e $\sqsupset$ é dado pela primeira coluna de cada matriz, e o segundo argumento vem dado pela primeira linha de cada matriz; por exemplo, $\sqsupset(\frac{1}{2}, 1) = 1$.) A noção de conseqüência semântica $\models_*$ é definida como segue: $\Gamma \models_* \varphi$ se, para toda valoração* v, se $v(\gamma) = 1$ para toda $\gamma \in \Gamma$, então $v(\varphi) = 1$.

Note que a tabela-verdade para a negação $-p := \sqsupset(p, 0)$ é a seguinte:

p	$-p$
1	0
$^1/_2$	1
0	1

Proposição 3.6.8. *O sistema P^+ é correto com relação à semântica de valorações* da Definição 3.6.7.*

Demonstração: Uma verificação rotineira, que deixamos como exercício para o leitor. $\square$

Proposição 3.6.9. *O sistema P^+ não é completo com relação à semântica de valorações clássicas.*

Demonstração: A fórmula $\varphi = (p_0 \Rightarrow p_0)$ é uma tautologia clássica, porém não é uma tautologia com relação à semântica de valorações* para P^+ introduzida na Definição 3.6.7. Com efeito, qualquer valoração* v tal que $v(p_0) = \frac{1}{2}$ produz $v(\varphi) = \frac{1}{2}$, portanto $\not\models_* \varphi$. Pela Proposição 3.6.8 temos que $\nvdash_{\mathsf{P}^+} \varphi$. Logo, P^+ não é uma axiomática completa para a LPC. $\square$

Como conseqüência imediata da prova da Proposição 3.6.9 obtemos que P^+ não satisfaz o Teorema da Dedução: com efeito, temos que $p_0 \vdash_{\mathsf{P}^+} p_0$, porém $\nvdash_{\mathsf{P}^+} (p_0 \Rightarrow p_0)$. O mesmo fenômeno acontece com a relação de conseqüência semântica $\models_*$.

3.7 Exercícios

1. Prove o Teorema 3.2.6.

2. Prove o Lema 3.2.8.

3. Seja S um sistema axiomático definido sobre uma assinatura C que contém (entre outras coisas) um conectivo binário $c \in C_2$. Suponha que existe uma relação de conseqüência semântica $\models$ determinada por um conjunto de valorações $v : L(C) \to \mathbf{2}$ tal que as seguintes propriedades são satisfeitas, para todo conjunto $\Gamma \cup \{\varphi, \psi\} \subseteq L(C)$:

   (a) Se $\Gamma \models \varphi$ então existe $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \models \varphi$;

(b) Se $\Gamma, \psi \models \varphi$ então $\Gamma \models c(\psi, \varphi)$;

(c) Se $\Gamma \vdash_{\mathsf{S}} c(\psi, \varphi)$ então $\Gamma, \psi \vdash_{\mathsf{S}} \varphi$;

(d) Se $\models \varphi$ então $\vdash_{\mathsf{S}} \varphi$.

(i) Prove que S satisfaz a completude forte: se $\Gamma \models \varphi$ então $\Gamma \vdash_{\mathsf{S}} \varphi$.
(ii) Enuncie condições análogas para provar a correção forte de S a partir da correção fraca de S. Usando essas condições, demonstre a correção forte de S a partir da correção fraca de S.

4. Mostre que os axiomas do sistema M são tautologias, e que a regra de Modus Ponens preserva tautologias. Prove a seguir o Teorema da Correção Fraca de M.

5. Usando o sistema M prove, para cada fórmula φ, ψ e γ, o seguinte:

   (a) $\vdash \varphi \Rightarrow \varphi$

   (b) $\vdash (\neg\varphi \Rightarrow \varphi) \Rightarrow \varphi$

   (c) $\varphi \Rightarrow \psi, \psi \Rightarrow \gamma \vdash \varphi \Rightarrow \gamma$

   (d) $\varphi \Rightarrow (\psi \Rightarrow \gamma) \vdash \psi \Rightarrow (\varphi \Rightarrow \gamma)$

   (e) $\vdash (\varphi \Rightarrow \psi) \Leftrightarrow (\neg\psi \Rightarrow \neg\varphi)$

   (f) $\vdash \neg\neg\psi \Leftrightarrow \psi$

   (g) $\vdash \varphi \Rightarrow (\neg\psi \Rightarrow \neg(\varphi \Rightarrow \psi))$

   (h) $\vdash (\varphi \Rightarrow \psi) \Rightarrow ((\neg\varphi \Rightarrow \psi) \Rightarrow \psi)$

   (i) $\vdash \varphi \Rightarrow (\varphi \vee \psi)$

   (j) $\vdash (\varphi \wedge \psi) \Rightarrow \varphi$

   (k) $\vdash \varphi \Rightarrow (\psi \Rightarrow (\varphi \wedge \psi))$

6. Mostre que os sistemas PC e CR são equivalentes.

7. Mostre que no sistema PC todos os axiomas e regras são independentes (i.e., retirando-se algum axioma ou regra o sistema não é mais completo).
   Sugestão: use tabelas trivalentes "esdrúxulas" que modelem todas as regras menos uma delas.

8. Considere o sistema axiomático M_1 escrito na linguagem Prop_1, formado pelos seguintes axiomas e regras de inferência:

(Axioma 1)	$\varphi \Rightarrow (\psi \Rightarrow \varphi)$
(Axioma 2)	$(\varphi \Rightarrow (\psi \Rightarrow \gamma)) \Rightarrow ((\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \gamma))$
(Axioma 3)	$\neg\neg\varphi \Rightarrow \varphi$
(Modus Ponens)	$\dfrac{\varphi \quad \varphi \Rightarrow \psi}{\psi}$

Provar que M_1 é correto com relação à semântica de valorações clássicas, mas não é completo.
Sugestão: use tabelas trivalentes que validem os axiomas e a regra de inferência de M_1, mas que não validem a fórmula $\varphi \Rightarrow \neg\neg\varphi$.

9. A partir das axiomas do sistema PC prove as seguintes propriedades da relação de conseqüência sintática (a relação de conseqüência que satisfaz estas propriedades é uma simplificação do *Método de Dedução Natural* que estudaremos no Capítulo 4):

Regras de introdução de conectivos:

(a) Se $\Gamma, \varphi \vdash \psi$ então $\Gamma \vdash \varphi \Rightarrow \psi$
(b) $\varphi, \psi \vdash \varphi \wedge \psi$
(c) $\varphi \vdash \varphi \vee \psi$ e $\psi \vdash \varphi \vee \psi$
(d) Se $\Gamma, \varphi \vdash \psi$ e $\Gamma, \varphi \vdash \neg\psi$ então $\Gamma \vdash \neg\varphi$

Regras de eliminação de conectivos:

(a) $\varphi, \varphi \Rightarrow \psi \vdash \psi$
(b) $\varphi \wedge \psi \vdash \varphi$ e $\varphi \wedge \psi \vdash \psi$
(c) Se $\Gamma, \varphi \vdash \gamma$ e $\Gamma, \psi \vdash \gamma$ então $\Gamma, \varphi \vee \psi \vdash \gamma$
(d) $\neg\neg\varphi \vdash \varphi$

10. Um *operador de conseqüência* de Tarski é uma operação entre conjuntos de fórmulas $Cn : \wp(L(C)) \to \wp(L(C))$ satisfazendo as seguintes propriedades:

(a) $X \subseteq Cn(X)$;
(b) $X \subseteq Y$ implica $Cn(X) \subseteq Cn(Y)$;
(c) $Cn(Cn(X)) = Cn(X)$.

Mostre que $Cn(\Gamma) = \{\varphi \in L(C) \ : \ \Gamma \vdash_{\mathsf{S}} \varphi\}$ é um operador de conseqüência, para todo sistema axiomático S. (O leitor interessado no estudo da lógica através de operadores de conseqüência pode consultar o livro [14].)

11. Demonstre o Teorema da Eliminação do corte para CR.
Sugestão: use indução na complexidade das demonstrações. Suponha que a regra $(Corte)$, na forma

$$\frac{\Gamma, \varphi \qquad \Gamma, \neg\varphi}{\Gamma},$$

foi a última a ser usada numa demonstração. Mostre então, por indução na complexidade de φ, que $(Corte)$ pode ser eliminada:
(i) Para φ atômico, conclua que $\varphi \in \Gamma$ e $\neg\varphi \in \Gamma$ logo $(Corte)$ é desnecessária.
(ii) Para φ da forma $\neg\psi$, use a hipótese de indução.
(iii) Para φ da forma $\varphi_0 \vee \varphi_1$ mostre que se $\vdash_{\mathsf{cr}} \Gamma, \neg(\varphi_0 \vee \varphi_1)$ então $\vdash_{\mathsf{cr}} \Gamma, \neg\varphi_0$ e $\vdash_{\mathsf{cr}} \Gamma, \neg\varphi_1$. A partir daí, use a hipótese de indução.

12. Prove o Teorema da Interpolação 3.5.3.
Sugestão: Mostre que cada item a seguir é válido e que o procedimento resolve o problema:

   (a) Pelas hipóteses do teorema, temos que $Var(\varphi_1) \cap Var(\varphi_2) \neq \emptyset$ (use a Proposição 2.2.4).

   (b) Tome $\Gamma = \{\psi \ : \ Var(\psi) \subseteq Var(\varphi_1) \cap Var(\varphi_2) \ \text{ e } \ \models \varphi_1 \Rightarrow \psi\}$. Então Γ é consistente.

   (c) Temos que $\Gamma \models \varphi_2$ (use o fato que, se $\Gamma \not\models \varphi_2$ então $\Gamma \cup \{\neg\varphi_2\}$ é consistente, e daí derive uma contradição a partir das hipóteses).

   (d) Existe Γ_0 finito contido em Γ tal que $\Gamma_0 \models \varphi_2$.

   (e) $\varphi = \bigwedge \Gamma_0$ (i.e., a conjunção dos elementos do conjunto finito Γ_0) satisfaz a conclusão do teorema.

13. Prove a Proposição 3.6.8.

Capítulo 4

Outros Métodos de Prova

Analisaremos neste capítulo três métodos alternativos de prova em sistemas formais. O primeiro deles, o método de *tablôs*, está baseado na *análise* das fórmulas (ao contrário do método axiomático introduzido anteriormente, no qual cada prova é uma *síntese* obtida de provas anteriores).

O método analítico tem vantagens do ponto de vista metamatemático, às quais nos referiremos, embora seja equivalente ao método axiomático tradicional conhecido como "hilbertiano".

O segundo método, denominado *dedução natural*, está baseado em regras de introdução e de eliminação de conectivos. A elaboração de provas seguindo estas regras visa simular, de certa maneira, o raciocínio de um matemático ao provar teoremas, dai o nome do método. Este método resulta ser equivalente ao método axiomático.

Finalmente descreveremos o método de *sequentes*, formado também por regras de introdução e de eliminação de conectivos, e cujos objetos, os *sequentes*, são pares de conjuntos de fórmulas no lugar de fórmulas. O método é equivalente ao método axiomático.

Desta maneira, neste capítulo são apresentadas três alternativas de método de prova para a lógica proposicional clássica, sendo que apenas o método de tablôs será estudado com algum detalhe.

As definições notacionais relativas a conjuntos de fórmulas (por exemplo, Γ, φ denotando o conjunto $\Gamma \cup \{\varphi\}$, etc.) introduzidas anteriormente, aplicam-se também neste capítulo.

4.1 O Método de Tablôs

Introduzimos nesta seção o método de tablôs,[1] introduzido por Beth (cf. [1]) e aperfeiçoado por Hintikka (cf. [7]) e Smullyan (cf. [13]). Como foi mencionado anteriormente, este é um método analítico, isto é, baseia-se na análise ou decomposição das fórmulas, em contraposição ao método axiomático (do tipo sintético) introduzido no capítulo anterior.

4.1.1 Descrição do método

Esta subseção é dedicada a descrever o método de tablôs, introduzindo as regras do sistema e os conceitos básicos para analisar o método. A linguagem proposicional que utilizaremos para o cálculo de tablôs é Prop.

Definição 4.1.1. Introduzimos as seguintes regras de análise:

$$(R_\vee) \quad \frac{p_0 \vee p_1}{p_0 \mid p_1}$$

$$(R_{\neg\vee}) \quad \frac{\neg(p_0 \vee p_1)}{\neg p_0, \neg p_1}$$

$$(R_{\neg\neg}) \quad \frac{\neg\neg p_0}{p_0}$$

Chamamos *configuração* a um conjunto finito de conjuntos de fórmulas. Seja Σ um conjunto de fórmulas (finito ou não). O *resultado da aplicação de uma regra de análise a* Σ é uma configuração $\{\Delta\}$ (no caso de $R_{\neg\vee}$ e $R_{\neg\neg}$) ou uma configuração $\{\Delta_1, \Delta_2\}$ (no caso de $R_\vee$), obtida da seguinte maneira:

- se Σ é da forma $\Gamma, \varphi \vee \psi$ então $\triangle_1$ é Γ, φ e $\triangle_2$ é Γ, ψ (em $R_\vee$);
- se Σ é da forma $\Gamma, \neg(\varphi \vee \psi)$ então $\triangle$ é da forma $\Gamma, \neg\varphi, \neg\psi$ (em $R_{\neg\vee}$);
- se Σ é da forma $\Gamma, \neg\neg\varphi$ então $\triangle$ é da forma Γ, φ (em $R_{\neg\neg}$).

Note que, dado um conjunto Σ, é possível aplicar diferentes regras nele, assim como aplicar a mesma regra de maneiras diferentes. Por exemplo, se $\Sigma = \{\varphi \vee \psi, \neg(\gamma \vee \delta), \gamma' \vee \delta'\}$, então podemos aplicar em Γ a regra $R_\vee$ de

[1] Adotaremos os neologismos "tablô" e "tablôs" para os termos franceses *tableau* e *tableaux*, respectivamente.

duas maneiras diferentes: aplicá-la com relação a $\varphi \vee \psi$ ou com relação a $\gamma' \vee \delta'$. Também podemos aplicar a regra $R_{\neg\vee}$ com relação a $\neg(\gamma \vee \delta)$.

Seja $\mathsf{C} = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_n\}$ uma configuração. O *resultado da aplicação de uma regra de análise a* C é uma nova configuração $\mathsf{C}' = (\mathsf{C} - \{\Sigma_i\}) \cup \{\Delta_{i_1}, \Delta_{i_2}\}$ (i_1 e i_2 possivelmente iguais) onde $\{\Delta_{i_1}, \Delta_{i_2}\}$ é o resultado da aplicação de alguma regra de análise a Σ_i (para algum $i \leq n$).

Um *tablô* é uma *seqüência* $\{\mathsf{C}_n\}_{n\in\mathbb{N}}$ (possivelmente estacionária) de configurações tal que $\mathsf{C}_{n+1} = \mathsf{C}_n$ ou C_{n+1} é obtida de C_n por aplicação de alguma regra de análise, para todo $n \in \mathbb{N}$.

Um *tablô para um conjunto* Σ é um tablô $\{\mathsf{C}_n\}_{n\in\mathbb{N}}$ tal que $\mathsf{C}_1 = \{\Sigma\}$.

Um conjunto de fórmulas Σ é *fechado* se existe uma fórmula φ tal que φ e $\neg\varphi$ pertencem a Σ; uma *configuração é fechada* se todos seus elementos o forem, e um *tablô* $\{\mathsf{C}_n\}_{n\in\mathbb{N}}$ *é fechado* se existe $n \in \mathbb{N}$ tal que $\mathsf{C}_m = \mathsf{C}_n$ para todo $m \geq n$, e C_n é fechada.

O caso não-fechado (em qualquer dos itens acima) é dito *aberto*.

Finalmente, dizemos que um tablô *está terminado* (ou *é completo*) se uma das três possibilidades acontece:

1. o tablô é fechado; ou

2. o tablô e aberto, não estacionário (isto é, para todo n existe $m > n$ tal que $\mathsf{C}_m \neq \mathsf{C}_n$; ou

3. o tablô é aberto, estacionário (isto é, existe n tal que, para todo $m > n$, $\mathsf{C}_m = \mathsf{C}_n$), e não é possível aplicar uma regra na configuração C_n.[2]

Observe que, se Σ é um conjunto finito de fórmulas, então todo tablô terminado para Σ é finito, isto é, deve terminar (aberto ou fechado) em finitos passos, sendo sempre estacionário. No caso da lógica de primeira ordem, veremos que nem sempre este é o caso.

Para facilidade de notação, podemos denotar tablôs utilizando árvores. Introduzimos a seguir os conceitos relativos a árvores.

Definição 4.1.2. Uma *árvore* (enraizada) é uma estrutura $\mathcal{A} = \langle |\mathcal{A}|, \mathsf{R}\rangle$ tal que $|\mathcal{A}|$ é um conjunto não vazio, e $\mathsf{R} \subseteq |\mathcal{A}| \times |\mathcal{A}|$ é uma relação satisfazendo as propriedades seguintes:

1. Existe um único $r \in |\mathcal{A}|$ tal que

   (a) não existe $x \in |\mathcal{A}|$ tal que $x\mathsf{R}r$,[3]

[2] Isto significa que os elementos abertos de C_n são conjuntos de literais.

[3] Se $\langle x, y\rangle \in \mathsf{R}$ escreveremos $x\mathsf{R}y$.

(b) se $y \in |\mathcal{A}|$, $y \neq r$, então existe um único $z \in |\mathcal{A}|$ tal que $z\mathsf{R}y$.

2. R não possui ciclos, isto é, não existem $x_1, \ldots, x_n$ em $|\mathcal{A}|$ tais que

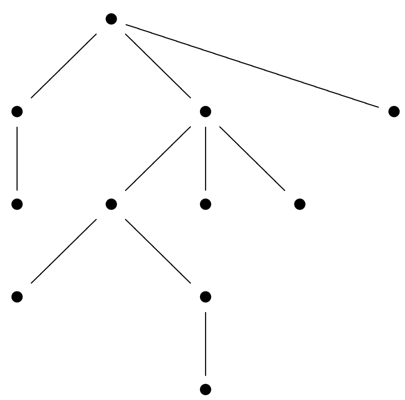
$$x_1\mathsf{R}x_2\mathsf{R}\cdots\mathsf{R}x_n\mathsf{R}x_1.$$

3. R não possui descensos infinitos, isto é, não existe uma sequência $\{x_n\}_{n\in\mathbb{N}}$ em $|\mathcal{A}|$ tal que

$$\cdots x_n\mathsf{R}x_{n-1}\mathsf{R}\cdots\mathsf{R}x_2\mathsf{R}x_1.$$

Os elementos de $|\mathcal{A}|$ são chamados de *nós*, e o nó r da primeira cláusula é chamado de *raiz.* Se $x\mathsf{R}y$ diremos que y é um *sucessor* de x, e x é o *predecessor* de y. Um nó sem sucessores é dito um *nó terminal.* Uma árvore é *finitamente gerada* se cada nó tem finitos sucessores. Se cada nó tem no máximo dois sucessores, a árvore é dita *diádica.*

A figura a seguir é uma representação de uma árvore finitamente gerada, não diádica.



Definição 4.1.3. Seja $\mathcal{A} = \langle|\mathcal{A}|, \mathsf{R}\rangle$ uma árvore com nó raiz r.

1. Uma seqüência finita $x_1 \cdots x_n$ em $|\mathcal{A}|$ é um *ramo finito* de $\mathcal{A}$ se $x_1 = r$, x_n é terminal e $x_i\mathsf{R}x_{i+1}$ para todo $1 \leq i \leq n-1$.

2. Uma seqüência infinita $\{x_n\}_{n\in\mathbb{N}}$ em $|\mathcal{A}|$ é um *ramo infinito* de $\mathcal{A}$ se $x_1 = r$ e $x_i\mathsf{R}x_{i+1}$ para todo $i \geq 1$.

3. Um *ramo* de $\mathcal{A}$ é um ramo finito ou infinito de $\mathcal{A}$.

Observação 4.1.4. Um tablô pode ser representado por árvores diádicas. Assim, cada nó é um conjunto de fórmulas, e as regras de análise produzem sucessores da maneira seguinte: $(R_\vee)$ produz dois sucessores,

$$\begin{tikzcd} & \Gamma, \varphi \vee \psi \arrow[dl, dash] \arrow[dr, dash] & \\ \Gamma, \varphi & & \Gamma, \psi \end{tikzcd}$$

enquanto que $(R_{\neg\vee})$ e $(R_{\neg\neg})$ produzem um único sucessor, respectivamente.

$$\begin{tikzcd} \Gamma, \neg(\varphi \vee \psi) \arrow[d, dash] & & \Gamma, \neg\neg\varphi \arrow[d, dash] \\ \Gamma, \neg\varphi, \neg\psi & & \Gamma, \varphi \end{tikzcd}$$

É fácil ver que existe uma correspondência biunívoca entre seqüências de árvores construídas de acordo com as regras de análise e tablôs.

(i) Dado um tablô $\mathsf{C}_1\mathsf{C}_2 \cdots \mathsf{C}_n \cdots$ definimos uma seqüência de árvores

$$\mathcal{A}_1\mathcal{A}_2 \cdots \mathcal{A}_n \cdots$$

da maneira seguinte:

1. $\mathcal{A}_1$ é a árvore com apenas um nó (raiz) Σ, se $\mathsf{C}_1 = \{\Sigma\}$. Note que os elementos da configuração C_1 são os nós terminais da árvore $\mathcal{A}_1$.

2. Suponha que a árvore $\mathcal{A}_n$ foi construída a partir da configuração C_n tal que os nós terminais de $\mathcal{A}_n$ são exatamente os elementos de C_n.

    (a) Se C_{n+1} é obtido de C_n substituindo um elemento Σ' de C_n pelo elemento Δ (resultado da aplicação da regra $(R_{\neg\vee})$ ou $(R_{\neg\neg})$) então definimos a árvore $\mathcal{A}_{n+1}$ como sendo a árvore $\mathcal{A}_n$ com o acréscimo do nó Δ como sucessor de Σ'. Note que os elementos da configuração C_{n+1} são os nós terminais da árvore $\mathcal{A}_{n+1}$.

    (b) Se C_{n+1} é obtido de C_n substituindo um elemento Σ' de C_n pelos elementos Δ_1, Δ_2 (resultado da aplicação da regra $(R_\vee)$) então definimos a árvore $\mathcal{A}_{n+1}$ como sendo a árvore $\mathcal{A}_n$ com o acréscimo dos nós Δ_1 e Δ_2 como sucessores de Σ'. Note que os elementos da configuração C_{n+1} são os nós terminais da árvore $\mathcal{A}_{n+1}$.

(ii) Considere uma seqüência de árvores $\mathcal{A}_1\mathcal{A}_2\cdots\mathcal{A}_n\cdots$ cujos nós são conjuntos de fórmulas tal que $\mathcal{A}_1$ consiste de um único nó Σ e, para cada n, a árvore $\mathcal{A}_{n+1}$ é obtida da árvore $\mathcal{A}_n$ pela aplicação de uma regra de análise, como indicado na Observação 4.1.4. Definimos um tablô

$$\mathsf{C}_1\mathsf{C}_2\cdots\mathsf{C}_n\cdots$$

da maneira seguinte:

1. $\mathsf{C}_1 = \{\Sigma\}$. Note que os elementos da configuração C_1 são os nós terminais da árvore $\mathcal{A}_1$.

2. Suponha que $\mathsf{C}_1\mathsf{C}_2\cdots\mathsf{C}_n$ é um tablô que foi definido a partir da seqüência de árvores $\mathcal{A}_1\mathcal{A}_2\cdots\mathcal{A}_n$ tal que os nós terminais de $\mathcal{A}_i$ são exatamente os elementos de C_i $(1 \leq i \leq n)$. Definimos C_{n+1} a partir de $\mathcal{A}_{n+1}$ da maneira seguinte:

   (a) Se aplicamos $(R_{\neg\vee})$ ou $(R_{\neg\neg})$ num nó terminal Σ' de $\mathcal{A}_n$ obtendo um sucessor Δ de Σ' em $\mathcal{A}_{n+1}$, então definimos C_{n+1} como sendo a configuração obtida de C_n substituindo Σ' por Δ. Note que os nós terminais de $\mathcal{A}_{n+1}$ são exatamente os elementos de C_{n+1}.

   (b) Se aplicamos $(R_\vee)$ num nó terminal Σ' de $\mathcal{A}_n$ obtendo dois sucessores Δ_1e Δ_2 de Σ' em $\mathcal{A}_{n+1}$, então definimos C_{n+1} como sendo a configuração obtida de C_n substituindo Σ' por Δ_1 e Δ_2. Note que os nós terminais de $\mathcal{A}_{n+1}$ são exatamente os elementos de C_{n+1}.

Por exemplo, considere o seguinte tablô $\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3$ e a seqüência de árvores correspondente:

$\Sigma_1 = \Gamma, \neg(\varphi \vee \psi) \vee \varphi$ $\qquad$ $\mathsf{C}_1 = \{\Sigma_1\}$ $\qquad$ Σ_1

$\Sigma_2 = \Gamma, \neg(\varphi \vee \psi)$ $\quad$ $\Sigma_3 = \Gamma, \varphi$ $\quad$ $\mathsf{C}_2 = \{\Sigma_2, \Sigma_3\}$

$$\begin{tikzcd}[arrows=dash]
& \Sigma_1 \arrow[dl] \arrow[dr] & \\
\Sigma_2 & & \Sigma_3
\end{tikzcd}$$

$\Sigma_4 = \Gamma, \neg\varphi, \neg\psi$ $\qquad$ $\mathsf{C}_3 = \{\Sigma_4, \Sigma_3\}$

$$\begin{tikzcd}[arrows=dash]
& \Sigma_1 \arrow[dl] \arrow[dr] & \\
\Sigma_2 \arrow[d] & & \Sigma_3 \\
\Sigma_4 & &
\end{tikzcd}$$

Dada a equivalência entre tablôs e seqüências de árvores geradas pelas regras analíticas, podemos falar em *ramos abertos* e *ramos fechados* de um tablô estacionário terminado em, digamos, n passos. Se $\Sigma' \in \mathsf{C}_n$ é aberto, então dizemos que Σ' é um *ramo aberto* do tablô. Caso contrário, Σ' é um *ramo fechado* do tablô. Esta denominação é justificada se consideramos a árvore $\mathcal{A}_n$ associada a C_n: os elementos de C_n são os nós terminais (que caracterizam os ramos) da árvore $\mathcal{A}_n$.

Definição 4.1.5. Dizemos que Γ *deriva* φ *analiticamente* se existe um tablô fechado para o conjunto $\Gamma, \neg\varphi$; denotamos tal fato por $\Gamma \vdash_T \varphi$. Dizemos que φ é um *teorema obtido analiticamente* se existe um tablô fechado para $\{\neg\varphi\}$) (isto é, se φ se deriva do conjunto vazio).

Observe que o método de tablôs é um método de refutação: para tentar provar que a fórmula φ é válida, supomos que ela pode ser falsa, isto é, partimos de $\neg\varphi$. Analogamente, para provar que φ segue logicamente de Γ, supomos que é possível ter simultaneamente Γ e $\neg\varphi$.

Para esta noção de dedução valem também as propriedades da relação de derivabilidade clássica (correspondentes ao Teorema 3.2.6):

Teorema 4.1.6. *Temos as seguintes propriedades:*

(a) $\varphi \vdash_T \varphi$
(b) Se $\vdash_T \varphi$ *então* $\Gamma \vdash_T \varphi$
(c) Se $\varphi \in \Gamma$ *então* $\Gamma \vdash_T \varphi$
(d) Se $\Gamma \vdash_T \varphi$ *e* $\Gamma \subseteq \Delta$ *então* $\Delta \vdash_T \varphi$
(e) $\Gamma \vdash_T \varphi$ *see existe* $\Gamma_0 \subseteq \Gamma$, Γ_0 *finito, tal que* $\Gamma_0 \vdash_T \varphi$

Demonstração: (a) Existe um tablô fechado para $\{\varphi, \neg\varphi\}$, esquematizado como uma árvore de um só nó: $\{\varphi, \neg\varphi\}$.
(b) Se existe um tablô fechado para $\{\neg\varphi\}$, então existe um tablô fechado para $\Gamma \cup \{\neg\varphi\}$ (basta acrescentar Γ a cada nó da árvore de $\{\neg\varphi\}$).
(c) Idem (a).
(d) Idem (b).
(e) Se existe um tablô fechado para $\Gamma \cup \{\neg\varphi\}$, este tablô consiste numa árvore finita, onde as regras foram aplicadas um número finito de vezes. Basta selecionar as fórmulas nas quais as regras se aplicaram e temos um conjunto Γ_0 finito tal que $\Gamma_0 \subseteq \Gamma$ e $\Gamma_0 \vdash_T \varphi$. A recíproca usa a parte (d). $\square$

Daremos a seguir alguns exemplos de deduções analíticas, utilizando árvores para visualizar melhor as deduções. Denotamos os ramos fechados por $*$.

Exemplo 4.1.7. $\vdash_T \varphi \Rightarrow (\psi \Rightarrow \varphi)$. Iniciamos uma árvore com $\neg(\varphi \Rightarrow (\psi \Rightarrow \varphi))$ e mostramos que se produz um tablô fechado.

$$\begin{array}{c} \neg(\neg\varphi \vee (\neg\psi \vee \varphi)) \\ | \\ \neg\neg\varphi, \neg(\neg\psi \vee \varphi) \\ | \\ \neg\neg\varphi, \neg\neg\psi, \neg\varphi \\ | \\ * \end{array}$$

(o ramo fecha em virtude da ocorrência de $\neg\neg\varphi$ e $\neg\varphi$. $\triangle$

Exemplo 4.1.8. $\varphi \vee \psi, \neg\psi \vdash_T \varphi$.

$$
\begin{array}{ccc}
 & \varphi \vee \psi, \neg\psi, \neg\varphi & \\
\swarrow & & \searrow \\
\varphi, \neg\psi, \neg\varphi & & \psi, \neg\psi, \neg\varphi \\
| & & | \\
* & & *
\end{array}
$$

$\triangle$

Exemplo 4.1.9. $\varphi \Rightarrow \psi, \varphi \Rightarrow \neg\psi \vdash_T \neg\varphi$.

$$
\begin{array}{ccccc}
 & & \neg\varphi \vee \psi, \neg\varphi \vee \neg\psi, \neg\neg\varphi & & \\
 & \swarrow & & \searrow & \\
\neg\varphi, \neg\varphi \vee \neg\psi, \neg\neg\varphi & & & & \psi, \neg\varphi \vee \neg\psi, \neg\neg\varphi \\
| & & & \swarrow & \searrow \\
* & & \psi, \neg\varphi, \neg\neg\varphi & & \psi, \neg\psi, \neg\neg\varphi \\
 & & | & & | \\
 & & * & & *
\end{array}
$$

$\triangle$

Note que não demonstramos todas as propriedades do Teorema 3.2.6; por exemplo, para demonstrar que se $\Gamma \vdash_T \varphi$ e $\varphi \vdash_T \psi$ então $\Gamma \vdash_T \psi$ precisamos provar uma propriedade mais forte da relação $\vdash_T$, a saber, o análogo para tablôs da *regra de modus ponens*.

Na verdade, precisamos provar uma espécie de contraparte do famoso teorema de Eliminação do Corte de Gentzen (veja o Teorema 4.3.4), para que possamos *introduzir* a regra de corte no sistema analítico:

> *Existem tablôs fechados para* Γ, φ *e* $\Gamma, \neg\varphi$ *see existe um tablô fechado para* Γ.

Após a introdução desta propriedade, poderemos provar a completude do método analítico e sua equivalência com o método hilbertiano. A demonstração deste teorema não é simples, e será feita na próxima subseção.

Finalizamos esta subseção com o estudo de regras derivadas.

Definição 4.1.10. Seja $\Gamma \subseteq \mathsf{Prop}$. Dizemos que Γ é *T-inconsistente* se existe um tablô fechado para Γ.

Definição 4.1.11. Uma *regra derivada* é uma regra analítica de um dos seguintes tipos.
a) uma regra derivada tipo *bifurcação* consiste de dois pares da forma

$$\langle \Gamma \cup \{\varphi\}, \Gamma \cup \{\varphi_1\}\rangle \ \text{ e } \ \langle \Gamma \cup \{\varphi\}, \Gamma \cup \{\varphi_2\}\rangle$$

tais que Γ é finito, e:

1. φ_1 e φ_2 (não necessariamente distintas) são subfórmulas de complexidade estritamente menor do que a complexidade de φ, e

2. $\Gamma \cup \{\varphi, \neg\varphi_1, \neg\varphi_2\}$ é T-inconsistente.

Denotamos uma regra derivada desse tipo por

$$\frac{\Gamma, \varphi}{\Gamma, \varphi_1 |\ \Gamma, \varphi_2} \text{ se } \varphi_1 \neq \varphi_2, \text{ e } \frac{\Gamma, \varphi}{\Gamma, \varphi_1} \text{ caso contrário.}$$

b) Uma regra derivada do tipo *linear* é um par da forma $\langle \Gamma \cup \{\varphi\}, \Gamma \cup \{\varphi_1, \varphi_2\}\rangle$ onde Γ é finito, e:

1. φ_1 e φ_2 (não necessariamente distintas) são subfórmulas de complexidade estritamente menor do que a complexidade de φ, e

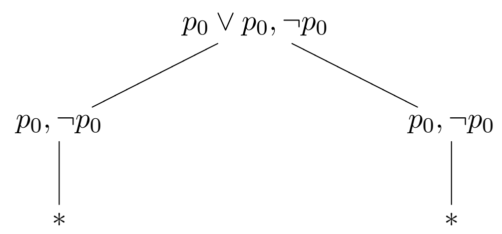
2. $\Gamma \cup \{\varphi, \neg\varphi_1\}$ e $\Gamma \cup \{\varphi, \neg\varphi_2\}$ são T-inconsistentes.

Denotamos uma regra desse tipo por

$$\frac{\Gamma, \varphi}{\Gamma, \varphi_1, \varphi_2} \text{ se } \varphi_1 \neq \varphi_2, \text{ e } \frac{\Gamma, \varphi}{\Gamma, \varphi_1} \text{ caso contrário.}$$

Exemplo 4.1.12. As seguintes são regras derivadas:

1. Todas as regras do sistema PC. Por exemplo $\dfrac{p_0 \vee p_0}{p_0}$ é uma regra derivada pois $\{p_0 \vee p_0\} \cup \{\neg p_0\}$ é T-inconsistente; de fato,



2. $\dfrac{p_0 \Rightarrow p_1}{\neg p_0 | p_1}$ porque $\{\neg p_0 \vee p_1, \neg\neg p_0, \neg p_1\}$ é T-inconsistente.

3. $\dfrac{\neg(p_0 \Rightarrow p_1)}{p_0, \neg p_1}$ (exercício).

4. $\dfrac{p_0 \wedge p_1}{p_0, p_1}$ (exercício).

5. $\dfrac{\neg(p_0 \wedge p_1)}{\neg p_0 | \neg p_1}$ (exercício).

$\triangle$

Podemos mostrar que as regras derivadas podem ser usadas livremente sem alterar o sistema analítico de provas.

Como deveriamos esperar, os sistemas sintético e analítico são equivalentes, no seguinte sentido:

Para todo $\Gamma \cup \{\varphi\}$, $\Gamma \vdash_T \varphi$ *see* $\Gamma \vdash_{\mathsf{PC}} \varphi$.

A prova dessa equivalência será feita na próxima subseção.

Acabamos de ver outra maneira, sintática, de verificar se φ é ou não um teorema, obtida diretamente do método analítico: pelo fato que os tablôs têm a chamada *propriedade da subfórmula*, isto é, cada regra produz somente subfórmulas das anteriores, e considerando que o número de subfórmulas de uma fórmula proposicional é finito, obtemos que *todo* tablô para um conjunto finito de fórmulas em Prop termina (fechado ou não) em finitos passos.[4] Se termina fechado, digamos para $\Gamma, \neg\varphi$, então $\Gamma \vdash_T \varphi$; caso contrário, $\Gamma \not\vdash_T \varphi$. Um dos grandes problemas da Teoria da Computação é saber se existem ou não métodos mais eficazes que estes dois métodos descritos acima. Tal questão, como veremos, está ligada ao problema $P \stackrel{?}{=} NP$, um dos mais difíceis problemas da computação teórica.

4.1.2 Equivalência do Sistema de Tablôs com o Sistema PC

Para provar a equivalência entre o método analítico e o método hilbertiano, precisamos primeiro provar o seguinte resultado, que estabelece uma espécie de recíproca do teorema de eliminação do corte dos sistemas de dedução natural.

[4]Veremos no Capítulo XXX que esta propriedade não é mais válida nos tablôs para a lógica de primeira ordem (dado que esta lógica é indecidível).

Teorema 4.1.13. *(Introdução do Corte) Os conjuntos* Γ, φ *e* $\Gamma, \neg\varphi$ *são* T*-inconsistentes see o conjunto* Γ *é* T*-inconsistente.*

Demonstração: ($\leftarrow$) Se existe um tablô fechado para Γ, é claro que existem tablôs fechados para quaisquer Δ, onde $\Gamma \subseteq \Delta$, e em particular para Γ, φ e $\Gamma, \neg\varphi$.

($\rightarrow$) Suponhamos que existam tablôs fechados para Γ, φ e $\Gamma, \neg\varphi$. Demonstramos então, por indução na soma das complexidades de φ e $\neg\varphi$, que existe também um tablô fechado para Γ.
Caso 1: φ é atômica. Dado que φ é atômica, então não existem regras analíticas aplicáveis a φ, portanto um tablô terminado para Γ é idêntico a um tablô terminado para Γ, φ, com a única exceção que φ serve para fechar todos os ramos abertos (se houver) do tablô para Γ (lembre que Γ, φ é T-inconsistente). Daqui inferimos que $\neg\varphi$ ocorre em todos os ramos abertos de um tablô terminado para Γ. Dado que $\neg\varphi$ também não pode ser utilizado nas regras analíticas (porque é um literal), e dado que $\Gamma, \neg\varphi$ também é T-inconsistente, pelo mesmo argumento vemos que φ (ou $\neg\neg\varphi$ e, *a posteriori*, φ) deve ocorrer em todo ramo aberto (se houver) de um tablô terminado para Γ. Ou seja, se houver um ramo aberto num tablô terminado para Γ, então o ramo deve conter simultaneamente φ e $\neg\varphi$, uma contradição. Portanto existe um tablô fechado para Γ (qualquer tablô terminado).
Caso 2: Se φ é da forma $\neg\psi$, e $\Gamma, \neg\psi$ e $\Gamma, \neg\neg\psi$ são ambos T-inconsistentes, então Γ, ψ é T-inconsistente, usando a regra $(R_{\neg\neg})$. Como $l(\psi) + l(\neg\psi) < l(\neg\psi) + l(\neg\neg\psi)$, por hipótese de indução concluímos que Γ é T-inconsistente.
Caso 3: Se φ é da forma $\psi \vee \gamma$, e $\Gamma, \psi \vee \gamma$ e $\Gamma, \neg(\psi \vee \gamma)$ são T-inconsistentes, então $\Gamma, \neg\psi, \neg\gamma$ é T-inconsistente (usando $(R_{\neg\vee})$) e Γ, ψ e Γ, γ são T-inconsistentes (usando $(R_{\vee})$). Pelo Teorema 4.1.6(d) inferimos que $\Gamma, \psi, \neg\gamma$ é T-inconsistente. Devido ao fato que $\Gamma, \neg\psi, \neg\gamma$ é T-inconsistente obtemos que $\Gamma, \neg\gamma$ é T-inconsistente, por hipótese de indução. Portanto, como $\Gamma, \neg\gamma$ e Γ, γ são T-inconsistentes, concluimos por hipótese de indução que Γ é T-inconsistente. $\square$

Com este resultado, podemos agora provar a primeira metade da equivalência entre os métodos de prova analítico e hilbertiano.

Teorema 4.1.14. *Para todo* $\Gamma \cup \{\varphi\} \subseteq \mathsf{Prop}$*, se* $\Gamma \vdash_{\mathsf{PC}} \varphi$ *então* $\Gamma \vdash_T \varphi$*.*

Demonstração: Por indução no comprimento n de uma prova em PC de φ a partir de Γ. Se $n = 1$ temos dois casos para analisar:
Caso 1: Se φ é uma instância $\neg\psi \vee \psi$ do axioma de PC. A árvore abaixo mostra que $\Gamma \cup \{\neg(\neg\psi \vee \psi)\}$ é T-inconsistente:

$$
\begin{array}{c}
\Gamma, \neg(\neg\psi \vee \psi) \\
| \\
\Gamma, \neg\neg\psi, \neg\psi \\
| \\
*
\end{array}
$$

Caso 2: Se $\varphi \in \Gamma$. Então $\Gamma \cup \{\neg\varphi\}$ é claramente T-inconsistente.

Suponha o resultado válido para toda fórmula que admite uma prova em PC a partir de Γ em n passos, e seja $\varphi_1 \cdots \varphi_{n+1} = \varphi$ uma prova em PC de φ a partir de Γ. Pela hipótese de indução, $\Gamma \vdash_T \varphi_i$ para todo $i \leq n$. Vamos mostrar que $\Gamma \vdash_T \varphi$. Devemos analisar um a um todos os casos que permitiram colocar φ_{n+1} (isto é, φ) na seqüência.
(a) Se φ é uma instância do axioma de PC, ou $\varphi \in \Gamma$. A prova é como acima.
(b) Se φ foi obtido de φ_i (para algum $1 \leq i \leq n$) pela regra de expansão (exp). Logo $\varphi = \varphi_i \vee \psi$ para alguma ψ. Por hipótese de indução $\Gamma, \neg\varphi_i$ é T-inconsistente; logo temos que $\Gamma, \neg\varphi_i, \neg\psi$ é T-inconsistente. Daí conclui-se que existe um tablô fechado para $\Gamma, \neg(\varphi_i \vee \psi)$, bastando aplicar a regra $(R_{\neg\vee})$ em primeiro lugar obtendo $\Gamma, \neg\varphi_i, \neg\psi$, e depois desenvolver um tablô fechado para esse conjunto.
(c) Se φ foi obtido de φ_i (para algum $1 \leq i \leq n$) pela regra de eliminação (elim). Então $\varphi_i = \varphi \vee \varphi$. Por hipótese de indução $\Gamma, \neg(\varphi \vee \varphi)$ é T-inconsistente, e pela regra $(R_{\neg\vee})$ $\Gamma, \neg\varphi$ é T-inconsistente.
(d) Se φ é obtido pela regra (assoc), a prova é análoga.
(e) Se φ é obtido de $\varphi_i = \psi \vee \gamma$, $\varphi_j = \neg\psi \vee \delta$ (para algum $i, j \leq n$) pela regra de corte, tal que $\varphi = \gamma \vee \delta$. Por hipótese de indução $\Gamma, \neg(\psi \vee \gamma)$ e $\Gamma, \neg(\neg\psi \vee \delta)$ são T-inconsistentes, logo $\Gamma, \neg\psi, \neg\gamma$ e $\Gamma, \neg\neg\psi, \neg\delta$ são T-inconsistentes. Daqui obtemos que $\Gamma, \neg\psi, \neg\gamma, \neg\delta$ e $\Gamma, \neg\neg\psi, \neg\gamma, \neg\delta$ são T-inconsistentes. Pelo Teorema 4.1.13, $\Gamma, \neg\gamma, \neg\delta$ é T-inconsistente, portanto $\Gamma, \neg(\gamma \vee \delta)$ é T-inconsistente. □

Para completar a prova da equivalência do método analítico com o método hilbertiano, precisamos de um resultado prévio.

Definição 4.1.15. Uma configuração $\mathsf{C} = \{\Gamma_1, \ldots, \Gamma_n\}$ é *satisfatível* se algum Γ_i foi satisfatível.

Lema 4.1.16. *Se Δ é T-inconsistente então Δ não é satisfatível (isto é, não existe uma valoração v tal que $v(\delta) = 1$ para toda $\delta \in \Delta$).*

Demonstração: Provaremos primeiro o seguinte
Fato: Se $\mathsf{C} = \{\Gamma_1, \ldots, \Gamma_n\}$ é uma configuração satisfatível, e C' obtêm-se de C por aplicação de alguma regra de tablô, então C' é satisfatível.

Com efeito, $\mathsf{C}' = \mathsf{C}'' \cup \{\Delta_1, \Delta_2\}$, com $\mathsf{C}'' = \mathsf{C} - \{\Gamma_i\}$ (para algum $1 \leq i \leq n$) e $\{\Delta_1, \Delta_2\}$ é o resultado de aplicar uma regra analítica a Γ_i (possivelmente $\Delta_1 = \Delta_2$). Seja v uma valoração que satisfaz algum elemento Γ_j de C. Se Γ_j pertence a C'' então v satisfaz C'. Caso contrário, v satisfaz Γ_i. Temos três casos para analisar, correspondentes às três regras analíticas:
Caso 1: Se Γ_i é $\Gamma, \varphi \vee \psi$. Então $\Delta_1 = \Gamma, \varphi$ e $\Delta_2 = \Gamma, \psi$. Neste caso, dado que $v(\varphi \vee \psi) = 1$ então $v(\varphi) = 1$ (em cujo caso v satisfaz Δ_1) ou $v(\psi) = 1$ (em cujo caso v satisfaz Δ_2). Em ambos os casos temos que v satisfaz C'.
Caso 2: Se Γ_i é $\Gamma, \neg(\varphi \vee \psi)$. Então $\Delta_1 = \Delta_2 = \Gamma, \neg\varphi, \neg\psi$. Dado que $v(\neg(\varphi \vee \psi)) = 1$ então $v(\varphi \vee \psi) = 0$, logo $v(\varphi) = v(\psi) = 0$ e então $v(\neg\varphi) = v(\neg\psi) = 1$. Dado que $v(\gamma) = 1$ para todo $\gamma \in \Gamma$ (pois v satisfaz Γ_i) então obtemos que v satisfaz Δ_1, logo v satisfaz C'.
Caso 3: Se Γ_i é $\Gamma, \neg\neg\varphi$. Então $\Delta_1 = \Delta_2 = \Gamma, \varphi$. Como antes, vemos que v satisfaz Δ_1, portanto v satisfaz C'. Isto prova o **Fato**.

Suponhamos então que Δ é T-inconsistente. Portanto existe um tablô fechado $\mathsf{C}_1\mathsf{C}_2 \cdots \mathsf{C}_n$ para Δ. Logo, $\mathsf{C}_1 = \{\Delta\}$ e C_{i+1} é obtido de C_i aplicando alguma regra de tablô ($1 \leq i \leq n-1$). Se Δ fosse satisfatível então C_1 seria satisfatível, portanto, pelo **Fato**, cada configuração do tablô fechado para Δ (incluíndo C_n) seria satisfatível, contradição. Isto mostra que Δ é insatisfatível. □

Teorema 4.1.17. *Para todo $\Gamma \cup \{\varphi\} \subseteq \mathsf{Prop}$, se $\Gamma \vdash_T \varphi$ então $\Gamma \vdash_{\mathsf{PC}} \varphi$.*

Demonstração: Se $\Gamma \vdash_T \varphi$ então $\Gamma \cup \{\neg\varphi\}$ é T-inconsistente. Logo, pelo lema anterior, $\Gamma \cup \{\neg\varphi\}$ é insatisfatível. Pela Proposição 3.4.10 temos que $\Gamma \models \varphi$, portanto $\Gamma \vdash_{\mathsf{PC}} \varphi$, pelo Teorema 3.4.6 de completude forte de PC. □

Obtemos finalmente a equivalência desejada entre tablôs e o sistema axiomático PC.

Teorema 4.1.18. *(Equivalência entre $\vdash_{\mathsf{PC}}$ e $\vdash_T$) Seja $\Gamma \cup \{\varphi\} \subseteq \mathsf{Prop}$. Então $\Gamma \vdash_{\mathsf{PC}} \varphi$ see $\Gamma \vdash_T \varphi$.*

A partir deste resultado, observamos que todas as propriedades de $\vdash_{\mathsf{PC}}$ são herdadas por $\vdash_T$.

Finalizamos esta seção observando que o método de tablôs nos fornece uma ferramenta para obter modelos de conjuntos satisfatíveis. Com efeito, basta apenas observar os elementos abertos da configuração final de um tablô terminado para um conjunto satisfatível Σ, obtendo facilmente todos os modelos de Σ. Começaremos pelo seguinte lema, cuja prova deixamos como exercício para o leitor:

Lema 4.1.19. *Seja C' uma configuração obtida da configuração C aplicando uma regra de análise. Logo, se C' é satisfatível por uma valoração v então C é também satisfatível pela valoração v.*

Vamos descrever a seguir um método para obter todos os modelos de um conjunto de fórmulas finito satisfatível Σ a partir de um tablô terminado para Σ. Seja $\mathsf{C}_1\mathsf{C}_2\cdots\mathsf{C}_n$ um tablô aberto terminado para Σ. Considere agora $\mathsf{C}_n = \{\Sigma_1, \ldots, \Sigma_k\}$. Como foi observado anteriormente, dado que o tablô está terminado, então $\Sigma_i = \{l^i_1, \ldots, l^i_{r_i}\}$, sendo que cada l^i_j é um literal para cada $j = 1, \ldots, r_i$ e $i = 1, \ldots, k$. Mais ainda, algum Σ_i deve ser aberto, porque o tablô é aberto. Sem perda de generalidade, suponhamos que $\Sigma_1, \ldots, \Sigma_s$, com $s \leq k$, são os ramos abertos de C_n. Veremos que cada Σ_i representa uma valoração $\bar{v}_i$ que satisfaz Σ ($i = 1, \ldots, s$). Mais ainda, toda valoração que satisfaz Σ é da forma $\bar{v}_i$ para algum $1 \leq i \leq s$. Fixemos $1 \leq i \leq s$, e seja P_i o conjunto de variáveis proposicionais que ocorrem em Σ_i. Definimos uma função $v_i : P_i \to \mathbf{2}$ tal que $v_i(p) = 1$ se $p \in \Sigma_i$, e $v_i(p) = 0$ se $\neg p \in \Sigma_i$. Dado que Σ_i é aberto, a função está bem definida. Seja $\bar{v}_i$ uma valoração obtida de v_i definindo valores arbitrários para cada variável proposicional p que não pertence a P_i. Logo $\bar{v}_i$ satisfaz Σ_i, portanto satisfaz C_n. Por indução em n e pelo Lema 4.1.19, provamos que $\bar{v}_i$ satisfaz Σ. Logo, cada Σ_i nos fornece um modelo (aliás, vários modelos, um para cada extensão $\bar{v}_i$ de v_i) de Σ ($i = 1, \ldots, s$). Veremos agora que *todo* modelo v de Σ é da forma $\bar{v}_i$ para algum $1 \leq i \leq s$. Para isso definimos, para cada $1 \leq i \leq s$, a fórmula $\psi_i = \neg \bigvee_{j=1}^{r_i} \neg l^i_j$ (lembre da notação estabelecida em 2.2.13). Observe que ψ_i representa toda valoração $\bar{v}_i$, no sentido seguinte: para cada valoração v, $v(\psi_i) = 1$ sse v coincide com v_i em P_i sse $v = \bar{v}_i$ (para alguma possível extensão $\bar{v}_i$ de v_i). Considere $\psi = \bigvee_{i=1}^{s} \psi_i$. Logo, para cada valoração v, $v(\psi) = 1$ sse existe $1 \leq i \leq s$ tal que $v = \bar{v}_i$ (para alguma possível extensão $\bar{v}_i$ de v_i). Provaremos agora que $\Sigma \models \psi$. Para isso, pelo Teorema 4.1.17 (e pela correção de PC) basta provar que existe um tablô fechado para $\Sigma, \neg\psi$. Para isso, considere o tablô $\{\mathsf{C}'_i\}_{i=1}^n$, onde C'_i é a configuração obtida de C_i acrescentando $\neg\psi$ como elemento de cada elemento de C_i ($i = 1, \ldots, n$). Isto é,

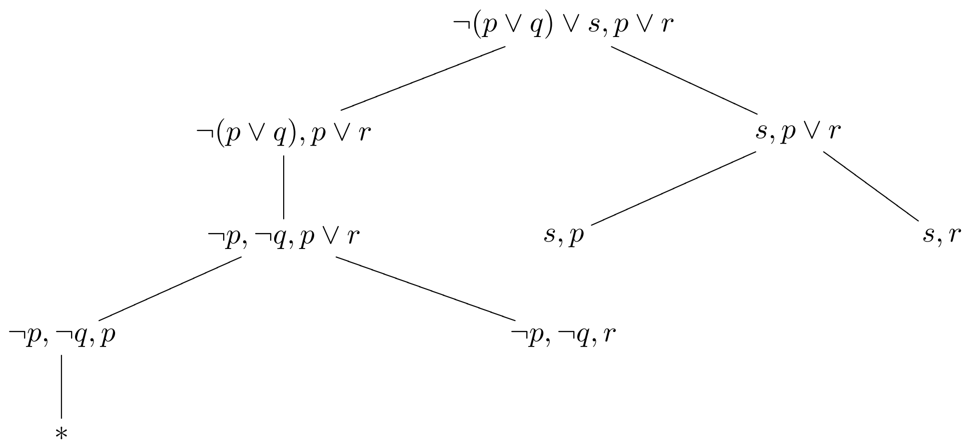
$$\mathsf{C}'_i = \{\Delta \cup \{\neg\psi\} \ : \ \Delta \in \mathsf{C}_i\}$$

para cada $1 \leq i \leq n$. A seqüência obtida é um tablô para $\Sigma, \neg\psi$. Continuando o tablô a partir da configuração C'_n, aplicando as regras apenas nos ramos abertos $\Sigma'_i = \Sigma_i, \neg\psi$ $(i = 1, \ldots, s)$ de C'_n, observamos que cada um destes ramos vai fechar após a aplicação das regras de análise (exercício para o leitor). Portanto, existe um tablô fechado para $\Sigma, \neg\psi$ e então $\Sigma \models \psi$. Logo, se uma valoração v satisfaz Σ então v satisfaz ψ, sendo portanto da forma $\bar{v}_i$ para algum $1 \leq i \leq s$.

A partir destes resultados, obtemos o seguinte:

Proposição 4.1.20. *Seja $\mathsf{C}_1\mathsf{C}_2 \cdots \mathsf{C}_n$ um tablô aberto terminado para um conjunto finito de fórmulas Σ. Então, podemos obter todos os modelos de Σ a partir de C_n.*

Exemplo 4.1.21. Seja $\Sigma = \{\neg(p \vee q) \vee s, p \vee r\}$, onde p, q, r, s são variáveis proposicionais. Considere o seguinte tablô terminado para Σ (exibimos apenas a árvore associada com a última configuração do tablô):



Considere v_1, v_2, v_3 tal que:

1. $v_1(p) = 0$, $v_1(q) = 0$, $v_1(r) = 1$;

2. $v_2(s) = 1$, $v_2(p) = 1$;

3. $v_3(s) = 1$, $v_3(r) = 1$.

A função v_1 fornece as valorações $\bar{v}_1^1$ e $\bar{v}_2^1$; a função v_2 fornece as valorações $\bar{v}_2^i$ $(i = 1, \ldots, 4)$; e a função v_3 fornece a valoração $\bar{v}_3$, definidas como segue:

	p	q	r	s
$\bar{v}_1^1$	0	0	1	1
$\bar{v}_1^2$	0	0	1	0
$\bar{v}_2^1$	1	1	1	1
$\bar{v}_2^2$	1	1	0	1
$\bar{v}_2^3$	1	0	1	1
$\bar{v}_2^4$	1	0	0	1
$\bar{v}_3$	0	1	1	1

Estas valorações são todos os modelos de Σ. $\triangle$

Graças à existência do método descrito acima para achar todos os modelos de um conjunto de fórmulas Γ a partir dos ramos abertos de um tablô terminado para Γ, e usando o Lema 4.1.16, obtemos o seguinte resultado, cuja prova deixamos como exercício para o leitor:

Corolário 4.1.22. *Seja Γ um conjunto finito de fórmulas. Então todo tablô terminado para Γ é aberto, ou todo tablô terminado para Γ é fechado.*

4.2 O Método de Dedução Natural

Nesta seção descreveremos brevemente o método de dedução natural. Este método, na sua forma atual, foi introduzido por Gentzen em 1934 (cf. [6]), se bem que existe um precedente de sistema de prova análogo, introduzido por Jáskowski em 1929 a partir de sugestões de Łukasiewics nos seus seminários em 1926 (cf. [8]).

A ideia do método é reproduzir os processos de inferência presentes no raciocínio intuitivo. Um bom exemplo destes processos são as demonstrações informais em matemática.

As regras do sistema são de dois tipos: de *introdução* de conectivos e de *eliminação* de conectivos; assim, cada conectivo (com exceção da constante **f**) possui ao menos uma regra de cada tipo, estipulando as circunstâncias em que esse conectivo pode ser introduzido ou eliminado. Uma característica importante destes sistemas é que cada derivação pode ser levada a uma derivação em *Forma Normal*, na qual não existem passos redundantes. Este teorema é equivalente ao conhecido teorema *Hauptsatz* (o teorema de Eliminação do Corte) estabelecido por Gentzen para o cálculo de sequentes. Os sistemas de dedução natural formam a base de uma importante área da lógica simbólica denominada *Teoria da prova* (*Proof Theory*). Nesta breve exposição apenas apresentaremos um sistema de dedução natural para a lógica proposicional clássica, explicaremos seu uso e ilustraremos o método com alguns exemplos. Recomendamos para o leitor interessado no tema a leitura do livro *Natural Deduction* de D. Prawitz ([11]), que transformou-se num clássico da área.

Definição 4.2.1. Considere a assinatura C^5 tal que $C^5_0 = \{\mathbf{f}\}$, $C^5_2 = \{\vee, \wedge, \Rightarrow\}$ e $C^5_n = \emptyset$ nos outros casos. Denotaremos o conjunto $L(C^5)$ por Sent. O sistema DN de dedução natural para a lógica proposicional clássica consiste das seguintes regras (como sempre, $\neg\varphi$ denota a fórmula $\varphi \Rightarrow \mathbf{f}$):

$$(\wedge I)\ \frac{\varphi \quad \psi}{\varphi \wedge \psi} \qquad (\wedge E1)\ \frac{\varphi \wedge \psi}{\varphi} \qquad (\wedge E2)\ \frac{\varphi \wedge \psi}{\psi}$$

$$(\vee I1)\ \frac{\varphi}{\varphi \vee \psi} \qquad (\vee I2)\ \frac{\psi}{\varphi \vee \psi} \qquad (\vee E)\ \frac{\varphi \vee \psi \quad \begin{matrix}[\varphi] \\ \gamma\end{matrix} \quad \begin{matrix}[\psi] \\ \gamma\end{matrix}}{\gamma}$$

$$(\Rightarrow I)\ \frac{\begin{matrix}[\varphi] \\ \psi\end{matrix}}{\varphi \Rightarrow \psi} \qquad (\Rightarrow E)\ \frac{\varphi \quad \varphi \Rightarrow \psi}{\psi}$$

$$(\mathbf{f}1)\ \frac{\mathbf{f}}{\varphi} \qquad (\mathbf{f}2)\ \frac{\begin{matrix}[\neg\varphi] \\ \mathbf{f}\end{matrix}}{\varphi}$$

Temos as seguintes restrições nas regras de $\mathbf{f}$: na regras ($\mathbf{f}1$) e ($\mathbf{f}2$) a fórmula φ é diferente de $\mathbf{f}$ e, na regra ($\mathbf{f}2$), a fórmula φ não é da forma $\neg\gamma$.

As restrições nas regras de **f** são inessenciais, e foram colocadas apenas para simplificar o estudo do sistema. Cada regra consiste de premissas (a parte de cima da barra) e de uma conclusão (a parte de baixo da barra). Partindo de um conjunto Γ de hipóteses, deduzimos conseqüências das fórmulas de Γ utilizando as regras, aplicando a seguir as regras nas conclusões, obtendo uma árvore invertida em que os nós terminais são as hipóteses utilizadas, e o nó raiz é a fórmula demonstrada. A notação

$$\begin{array}{c}[\varphi]\\ \psi\end{array}$$

utilizada nas regras ($\vee E$), ($\Rightarrow I$) e (**f**2) significa que estamos supondo a existência de uma derivação de ψ a partir de φ como uma de suas hipóteses. Nesse caso, após a aplicação da regra que utiliza

$$\begin{array}{c}[\varphi]\\ \psi\end{array}$$

entre suas premissas, algumas ocorrências (todas, algumas, nenhuma) da premissa φ *podem* ser eliminadas como hipóteses. Assim, o resultado (a conclusão da regra) não vai mais depender da premissa φ, caso todas as ocorrências de φ como premissa tenham sido eliminadas. Isto reflete a ideia do Teorema da Dedução: se $\Gamma, \varphi \vdash \psi$ (ψ depende de Γ, φ) então $\Gamma \vdash \varphi \Rightarrow \psi$ ($\varphi \Rightarrow \psi$ depende de Γ). Vamos analisar dois exemplos para esclarecer o método:

Exemplos 4.2.2.
(1) Provaremos que $(\varphi \vee \psi) \Rightarrow \gamma$ é deduzido em DN a partir do conjunto de hipóteses $\{\varphi \Rightarrow \gamma, \psi \Rightarrow \gamma\}$. Primeiro de tudo, a seguinte derivação

$$\frac{\varphi \Rightarrow \gamma \quad \varphi}{\gamma}$$

mostra que γ é deduzido em DN a partir de $\{\varphi \Rightarrow \gamma, \varphi\}$. Analogamente provamos que γ é deduzido em DN a partir de $\{\psi \Rightarrow \gamma, \psi\}$. Juntando estas duas derivações com a hipótese adicional $\varphi \vee \psi$ inferimos, aplicando a regra ($\vee E$), a fórmula γ, podendo descarregar as hipóteses φ e ψ:

$$\frac{\dfrac{\varphi \Rightarrow \gamma \quad [\varphi]}{\gamma} \qquad \dfrac{\psi \Rightarrow \gamma \quad [\psi]}{\gamma} \qquad \varphi \vee \psi}{\gamma}$$

Isto significa que γ é derivável em DN a partir de $\{\varphi \Rightarrow \gamma, \psi \Rightarrow \gamma, \varphi \vee \psi\}$. Finalmente, aplicando a regra ($\Rightarrow I$) podemos descarregar a hipótese $\varphi \vee \psi$, obtendo uma derivação de $(\varphi \vee \psi) \Rightarrow \gamma$ em DN a partir de $\{\varphi \Rightarrow \gamma, \psi \Rightarrow \gamma\}$.

$$\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \gamma \quad [\varphi]^1}{\gamma} \qquad \dfrac{\psi \Rightarrow \gamma \quad [\psi]^1}{\gamma} \qquad [\varphi \vee \psi]^2}{\gamma}\,{}^{1}}{(\varphi \vee \psi) \Rightarrow \gamma}\,{}^{2}$$

Os números (1 e 2) foram colocados para individualizar a regra pela qual foram descarregadas as hipóteses. Assim, na aplicação da regra 1 foram descarregadas as hipóteses marcadas com 1, e na aplicação da regra 2 foi descarregada a hipótese marcada com 2.

(2) Provaremos que $\gamma \Rightarrow (\varphi \wedge \psi)$ é deduzido em DN a partir da hipótese $(\gamma \Rightarrow \varphi) \wedge (\gamma \Rightarrow \psi)$. A seguinte derivação em DN prova esse fato:

$$\dfrac{\dfrac{\dfrac{[\gamma]^1 \quad \dfrac{(\gamma \Rightarrow \varphi) \wedge (\gamma \Rightarrow \psi)}{\gamma \Rightarrow \varphi}}{\varphi} \qquad \dfrac{[\gamma]^1 \quad \dfrac{(\gamma \Rightarrow \varphi) \wedge (\gamma \Rightarrow \psi)}{\gamma \Rightarrow \psi}}{\psi}}{\varphi \wedge \psi}}{\gamma \Rightarrow (\varphi \wedge \psi)}\,{}^{1}$$

(3) Provaremos que em DN φ é derivável a partir de $\neg\neg\varphi$, e vice-versa. Considere as seguintes derivações (lembrando que $\neg\neg\varphi$ é $\neg\varphi \Rightarrow \mathbf{f}$):

$$\dfrac{\dfrac{\neg\neg\varphi \quad [\neg\varphi]^1}{\mathbf{f}}}{\varphi}\,{}^{1} \qquad\qquad \dfrac{\dfrac{\varphi \quad [\neg\varphi]^1}{\mathbf{f}}}{\neg\neg\varphi}\,{}^{1}$$

(4) Provaremos que em DN a fórmula $\neg\varphi \vee \varphi$ é um teorema. Considere a seguinte derivação (lembrando que $\neg\varphi$ é $\varphi \Rightarrow \mathbf{f}$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\varphi]^1}{\neg\varphi \vee \varphi} \quad [\neg(\neg\varphi \vee \varphi)]^2}{\mathbf{f}}}{\neg\varphi}\,{}^{1}}{\neg\varphi \vee \varphi} \qquad [\neg(\neg\varphi \vee \varphi)]^2}{\mathbf{f}}}{\neg\varphi \vee \varphi}\,{}^{2}$$

Na derivação acima, o número 1 indica uma aplicação da regra $(\Rightarrow I)$, enquanto que o número 2 indica uma aplicação da regra ($\mathbf{f}$2). $\triangle$

O sistema DN é correto e completo para a semântica da lógica proposicional clássica. Isto é:

Teorema 4.2.3. *Seja* $\Gamma \cup \{\varphi\}$ *um conjunto finito de fórmulas em* Sent. *Então existe uma derivação em* DN *de* φ *a partir de* Γ *sse* $\Gamma \models \varphi$.

4.3 O Método de Sequentes

Finalizamos este capítulo com uma rápida descrição do método de sequentes. Este método foi introduzido por Gentzen mas, ao contrário do método de dedução natural, o cálculo de sequentes foi intoduzido por motivos puramente técnicos, não possuindo qualquer justificativa ou motivação filosófica. De fato, o cálculo de sequentes foi introduzido apenas como um formalismo apropriado para provar o Teorema de Eliminação e Corte ou *Haupsatz*:

> *The Hauptsatz says that every purely logical proof can be reduced to a definite, though not unique, normal form. Perhaps we may express the essential properties of such a normal proof by saying: it is not roundabout... In order to be able to prove the Hauptsatz in a convenient form, I had to provide a logical calculus especially for the purpose. For this the natural calculus proved unsuitable.*
>
> Gentzen, "Investigations into logical deduction"

Tanto o método de dedução natural quanto o método de sequentes são atualmente importantes ferramentas para o desenvolvimento e a apresentação de sistemas lógicos.

A característica principal do método de sequentes é a utilização nas provas de sequentes no lugar de fórmulas.

Definição 4.3.1. Seja C^6 a assinatura tal que $C^6_0 = \{\,\mathbf{f}, \mathbf{t}\,\}$, $C^6_1 = \{\neg\}$, $C^6_2 = \{\vee, \wedge, \Rightarrow\}$ e $C^6 = \emptyset$ nos outros casos. Um *sequente* (sobre C^6) é um par $\langle \Gamma, \Delta \rangle$ de conjuntos finitos de fórmulas não simultaneamente vazios. Um sequente $\langle \Gamma, \Delta \rangle$ será denotado por $\Gamma \Longrightarrow \Delta$.

Na assinatura C^6, a constante $\mathbf{t}$ representa o valor de verdade 1, enquanto que a constante $\mathbf{f}$ representa, como sempre, o valor de verdade 0.

Definição 4.3.2. O cálculo SQ de sequentes para a lógica proposicional clássica é dado pelas seguintes regras:

$$(Ax)\ \frac{}{\Gamma, \varphi \Longrightarrow \Delta, \varphi} \qquad (Corte)\ \frac{\Gamma \Longrightarrow \Delta, \varphi \quad \Gamma, \varphi \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta}$$

$$(\wedge E)\ \frac{\Gamma, \varphi, \psi \Longrightarrow \Delta}{\Gamma, \varphi \wedge \psi \Longrightarrow \Delta} \qquad (\wedge D)\ \frac{\Gamma \Longrightarrow \Delta, \varphi \quad \Gamma \Longrightarrow \Delta, \psi}{\Gamma \Longrightarrow \Delta, \varphi \wedge \psi}$$

$$(\Rightarrow E)\ \frac{\Gamma \Longrightarrow \Delta, \varphi \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \Rightarrow \psi \Longrightarrow \Delta} \qquad (\Rightarrow D)\ \frac{\Gamma, \varphi \Longrightarrow \Delta, \psi}{\Gamma \Longrightarrow \Delta, \varphi \Rightarrow \psi}$$

$$(\vee E)\ \frac{\Gamma, \varphi \Longrightarrow \Delta \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \vee \psi \Longrightarrow \Delta} \qquad (\vee D)\ \frac{\Gamma \Longrightarrow \Delta, \varphi, \psi}{\Gamma \Longrightarrow \Delta, \varphi \vee \psi}$$

$$(\neg E)\ \frac{\Gamma \Longrightarrow \Delta, \varphi}{\Gamma, \neg\varphi \Longrightarrow \Delta} \qquad (\neg D)\ \frac{\Gamma, \varphi \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta, \neg\varphi}$$

$$(\mathbf{f}E)\ \frac{}{\Gamma, \mathbf{f} \Longrightarrow \Delta} \qquad (\mathbf{t}D)\ \frac{}{\Gamma \Longrightarrow \Delta, \mathbf{t}}$$

Intuitivamente, um sequente $\gamma_1, \ldots, \gamma_n \Longrightarrow \delta_1, \ldots, \delta_k$ denota $\bigwedge_{i=1}^{n} \gamma_i \vdash \bigvee_{i=1}^{k} \delta_i$. Como no caso da dedução natural, as provas no cálculo SQ são árvores invertidas, cujos nós são sequentes. Os nós terminais são instâncias dos axiomas (Ax), ($\perp$E) ou ($\mathbf{t}$D), os predecessores são obtidos pela aplicação de alguma regra em SQ, e o nó raiz é o sequente a ser demonstrado. Podemos executar o proceso de derivação em sentido inverso (*backward*), começando pelo sequente a ser demonstrado e aplicando alguma regra que tenha como conclusão o sequente sendo analisado, obtendo como sucessor o antecedente da regra. Continuando com este processo, vemos que as derivações em SQ realizadas no sentido *backward* são árvores diádicas.

Exemplos 4.3.3.
(1) Provaremos o sequente $\varphi \wedge (\psi \vee \gamma) \Longrightarrow (\varphi \wedge \psi) \vee (\varphi \wedge \gamma)$ em SQ. Considere a seguinte derivação em SQ:

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi, \psi \Longrightarrow \varphi, \varphi \wedge \gamma \quad \varphi, \psi \Longrightarrow \psi, \varphi \wedge \gamma}{\varphi, \psi \Longrightarrow \varphi \wedge \psi, \varphi \wedge \gamma} \qquad \dfrac{\varphi, \gamma \Longrightarrow \varphi, \varphi \wedge \psi \quad \varphi, \gamma \Longrightarrow \gamma, \varphi \wedge \psi}{\varphi, \gamma \Longrightarrow \varphi \wedge \psi, \varphi \wedge \gamma}}{\varphi, \psi \vee \gamma \Longrightarrow \varphi \wedge \psi, \varphi \wedge \gamma}}{\varphi \wedge (\psi \vee \gamma) \Longrightarrow \varphi \wedge \psi, \varphi \wedge \gamma}}{\varphi \wedge (\psi \vee \gamma) \Longrightarrow (\varphi \wedge \psi) \vee (\varphi \wedge \gamma)}$$

(2) Provaremos a seguir o sequente $(\varphi \wedge \psi) \vee (\varphi \wedge \gamma) \Longrightarrow \varphi \wedge (\psi \vee \gamma)$ em SQ, mas agora executando as regras em sentido *backward*. Assim, começando pelo sequente a ser demonstrado geramos a seguinte árvore diádica:

$$\begin{array}{ccc}
 & (\varphi \wedge \psi) \vee (\varphi \wedge \gamma) \Longrightarrow \varphi \wedge (\psi \vee \gamma) & \\
 / & & \backslash \\
\mathcal{A}_1 & & \mathcal{A}_2
\end{array}$$

onde $\mathcal{A}_1$ e $\mathcal{A}_2$ são as seguintes árvores diádicas, respectivamente:

$$\begin{array}{ccc}
 & \varphi \wedge \psi \Longrightarrow \varphi \wedge (\psi \vee \gamma) & \\
 & | & \\
 & \varphi, \psi \Longrightarrow \varphi \wedge (\psi \vee \gamma) & \\
 / & & \backslash \\
\varphi, \psi \Longrightarrow \varphi & & \varphi, \psi \Longrightarrow \psi \vee \gamma \\
 & & | \\
 & & \varphi, \psi \Longrightarrow \psi, \gamma
\end{array}$$

$$
\begin{array}{ccc}
 & \varphi \wedge \gamma \Longrightarrow \varphi \wedge (\psi \vee \gamma) & \\
 & \varphi, \gamma \Longrightarrow \varphi \wedge (\psi \vee \gamma) & \\
\varphi, \gamma \Longrightarrow \varphi & & \varphi, \gamma \Longrightarrow \psi \vee \gamma \\
 & & \varphi, \gamma \Longrightarrow \psi, \gamma
\end{array}
$$

Evidentemente, invertendo a árvore de derivação acima obtemos uma derivação "genuina" em SQ. Note que poderiamos ter aplicado, no primeiro passo da construção da árvore, a regra ($\vee$R) no lugar da regra ($\wedge$E), obtendo outra árvore de derivação. $\triangle$

O célebre teorema *Hauptsatz* de Gentzen estabelece que a regra (Corte) é redundante e pode ser eliminada:

Teorema 4.3.4 (Hauptsatz)**.** *O sistema SQ^- obtido de SQ eliminando a regra (Corte) é equivalente com o sistema SQ.*

De fato, a regra (Corte) foi introduzida por Gentzen apenas para obter uma função que transforme derivações de DN em derivações de SQ. Prova-se então que o sistema SQ (ou, equivalentemente, SQ^-) é correto e completo para a semântica da lógica proposicional clássica, considerando valorações clássicas v tais que $v(\mathbf{t}) = 1$. Isto é:

Teorema 4.3.5. *Sejam $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ e $\Delta = \{\delta_1, \ldots, \delta_k\}$ dois conjuntos finitos de fórmulas em $L(C^6)$. Então existe uma derivação em SQ do sequente $\Gamma \Longrightarrow \Delta$ sse $\bigwedge_{i=1}^{n} \gamma_i \models \bigvee_{i=1}^{k} \delta_i$.*

Corolário 4.3.6. *Sejam $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ e $\Delta = \{\delta_1, \ldots, \delta_k\}$ dois conjuntos finitos de fórmulas em $L(C^6)$. Então existe uma derivação em SQ^- do sequente $\Gamma \Longrightarrow \Delta$ sse $\bigwedge_{i=1}^{n} \gamma_i \models \bigvee_{i=1}^{k} \delta_i$.*

Observe que o cálculo de sequentes sem cortes SQ^-, se executado em sentido inverso (*backward*) como no Exemplo 4.3.3, resulta ser um método de prova analítico, bastante semelhante ao método de tablôs.

4.4 Exercícios

1. Prove que as regras do Exemplo 4.1.12 são derivadas.

2. Prove o Lema 4.1.19.

3. Provar o Corolário 4.1.22.

4. Escolha uma fórmula arbitrária e calcule sua FND usando tablôs.

5. Provar o seguinte utilizando tablôs, dedução natural e sequentes (calcule algumas das derivações de sequentes no sentido *backward*):

   (a) $\varphi \vee (\psi \wedge \delta) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \delta)$;

   (b) $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$;

   (c) $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$.

6. Prove diretamente (sem usar os teoremas de adequação) que o método de tablôs simula o método de sequentes no sentido seguinte:

   se $\Gamma \Longrightarrow \delta_1, \ldots, \delta_n$ é um sequente demonstrável em SQ então o conjunto $\Gamma, \neg\delta_1, \ldots, \neg\delta_n$ é T-inconsistente.

Capítulo 5

Álgebras, Ordens, Reticulados e Semântica Algébrica

Neste capítulo estudaremos algumas ferramentas para o estudo da lógica formal, que tem interesse *per se*, sendo aplicáveis a diferentes ramos da matemática e da computação: álgebras abstratas (ou universais), teoria de ordem e reticulados. A aplicação que daremos destes conceitos para a lógica proposicional clássica consiste na definição da álgebra de Lindembaum-Tarski da LPC, uma semântica algébrica extremamente útil para analisar certas lógicas proposicionais.

5.1 Álgebras abstratas

Nesta seção definiremos alguns poucos conceitos básicos de álgebras abstratas (ou universais), necessários para discutir as noções sobre reticulados que estabeleceremos posteriormente. Uma excelente referência sobre álgebra universal é [2].

Notação 5.1.1. Dados conjuntos X e Y, então X^Y denotará o conjunto de todas as funções $f : Y \to X$.

Observe que $X^Y = \emptyset$ sse $Y \neq \emptyset$ e $X = \emptyset$. Se $Y = \emptyset$ então $X^Y = \{\emptyset\}$ (a única função que existe com domínio vazio e contra-domínio X é o conjunto vazio de pares ordenados).

Definição 5.1.2. Seja $C = \{C_n\}_{n\in\mathbb{N}}$ uma assinatura proposicional. Uma *álgebra abstrata de tipo* C (ou simplesmente uma *álgebra* de tipo C) é um par

$\mathsf{A} = \langle |\mathsf{A}|, F\rangle$ tal que $|\mathsf{A}|$ é um conjunto não vazio (o *domínio* da álgebra) e F é uma função $F : |C| \to \bigcup_{n\in\mathbb{N}} |\mathsf{A}|^{|\mathsf{A}|^n}$ tal que $F(c) : |\mathsf{A}|^n \to |\mathsf{A}|$ se $c \in C_n$. Se $c \in |C|$ então $F(c)$ será denotado por c^{A}. Uma álgebra A será frequentemente denotada como um par $\mathsf{A} = \langle |\mathsf{A}|, \{O_n\}_{n\in\mathbb{N}}\rangle$ tal que $O_n = \{c^{\mathsf{A}} \ : \ c \in C_n\}$ para todo $n \in \mathbb{N}$. Isto é, cada O_n é o conjunto dos chamados *operadores n-ários* de A. Definimos o conjunto de operadores de A como sendo o conjunto $|O| = \bigcup_{n\in\mathbb{N}} O_n$.

Exemplos 5.1.3.
(1) Se C é uma assinatura então $\mathsf{A} = \langle L(C), C\rangle$ é uma álgebra em que $c^{\mathsf{A}} = c$ para todo $c \in |C|$. Isto é, os operadores n-ários são os conectivos n-ários.
(2) Temos que $\mathcal{B}_\mathbf{2} = \langle \mathbf{2}, \{\sqcup, \sqcap, \sqsupset, \bot\}\rangle$ é uma álgebra de tipo C^5 (veja Definição 4.2.1). △

Note que no Exemplo 5.1.3(2) escrevemos o conjunto de operadores no lugar da família de operadores. Faremos isto com frequência, quando o número de operadores seja finito e a aridade de cada operador seja dada.

Definição 5.1.4. Duas álgebras $\mathsf{A}_i = \langle |\mathsf{A}_i|, F_i\rangle$ $(i = 1, 2)$ do mesmo tipo C são ditas *similares*. A assinatura C é o *tipo de similaridade* de A_i $(i = 1, 2)$.

Definição 5.1.5. Sejam $\mathsf{A}_i = \langle |\mathsf{A}_i|, F_i\rangle$ $(i = 1, 2)$ duas álgebras similares com tipo de similaridade C. Um *homomorfismo h de A_1 em A_2*, denotado $h : \mathsf{A}_1 \to \mathsf{A}_2$, é uma função $h : |\mathsf{A}_1| \to |\mathsf{A}_2|$ tal que $h(c^{\mathsf{A}_1}(a_1, \ldots, a_n)) = c^{\mathsf{A}_2}(h(a_1), \ldots, h(a_n))$ para todo $n \in \mathbb{N}$, $c \in C_n$ e $a_1, \ldots, a_n \in |\mathsf{A}_1|$.

Exemplo 5.1.6. Consideremos novamente a assinatura C^5 introduzida na Definição 4.2.1, lembrando que $|C^5| = \{\vee, \wedge, \Rightarrow, \mathbf{f}\}$ e $L(C^5)$ é denotado por Sent. Considere $\mathcal{B}_\mathbf{2}$ como álgebra de tipo de similaridade C^5 (Exemplo 5.1.3(2)). Lembremos que o conjunto $L(C^5)$ é denotado por Sent. Então uma valoração clássica $v : \mathsf{Sent} \to \mathbf{2}$ é um homomorfismo. Da mesma maneira podemos considerar as valorações clássicas $v : L(C^i) \to \mathbf{2}$ como sendo homomorfismos, dado que $\mathbf{2}$ pode ser vista como álgebra de tipo de similaridade C^i $(i = 1, \ldots, 6)$. △

Definição 5.1.7. Dada uma álgebra $\mathsf{A} = \langle |\mathsf{A}|, \{O_n\}_{n\in\mathbb{N}}\rangle$ de tipo C e $B \subseteq |\mathsf{A}|$ um conjunto não vazio tal que $c^{\mathsf{A}}(b_1, \ldots, b_n) \in B$ para todo $n \in \mathbb{N}$, $c \in C_n$ e $b_1, \ldots, b_n \in B$, então dizemos que B (com as operações restritas a B) é uma *subálgebra* de A.

Obviamente uma subálgebra de uma álgebra A de tipo C é uma álgebra de tipo C, portanto similar a A. Se $c \in C_n$ então $c^B = c^{\mathsf{A}}|_{B^n}$ tal que

$c^{\mathsf{A}}|_{B^n} : B^n \to B$ é dada por $c^{\mathsf{A}}|_{B^n}(b_1, \ldots, b_n) = c^{\mathsf{A}}(b_1, \ldots, b_n)$ para todo $b_1, \ldots, b_n \in B$. Note que, dada uma álgebra A, então a interseção dos domínios de uma família arbitrária de subálgebras de A é uma subálgebra de A (desde que a interseção seja um conjunto não vazio). Portanto, dado $X \subseteq |\mathsf{A}|$ não vazio, definimos *a álgebra gerada por* X como sendo a subálgebra $A(X)$ de A com domínio

$$|A(X)| = \bigcap\{|\mathsf{A}'| \ : \ \mathsf{A}' \text{ é uma subálgebra e } \mathsf{A} \text{ e } X \subseteq |\mathsf{A}'|\}.$$

Dizemos que X *gera* $A(X)$. Uma álgebra A é *gerada por* X se $\mathsf{A} = A(X)$.

Se A é gerada por X e, para toda álgebra A' similar a A e para toda função $h_0 : X \to |\mathsf{A}'|$ existe um único homomorfismo $h : \mathsf{A} \to \mathsf{A}'$ estendendo h_0 (isto é, tal que $h(x) = h_0(x)$ para todo $x \in X$), então dizemos que A é *livremente gerada por* X (com relação ao tipo de similaridade de A).

Exemplo 5.1.8. A álgebra $\langle L(C), C\rangle$ é livremente gerada por $\mathcal{V}$. △

Proposição 5.1.9. *Sejam* $\mathsf{A}_i = \langle|\mathsf{A}_i|, \{O^i_n\}_{n\in\mathbb{N}}\rangle$ *(*$i = 1, 2$*) duas álgebras similares, e sejam* $h_i : \mathsf{A}_1 \to \mathsf{A}_2$ *dois homomorfismos (*$i = 1, 2$*). Se* A_1 *é livremente gerada por* X *e* h_1 *e* h_2 *coincidem em* X *(isto é,* $h_1(x) = h_2(x)$ *para todo* $x \in X$*), então* $h_1 = h_2$*.*

Demonstração: Basta observar que $\{a \in |\mathsf{A}| \ : \ h_1(a) = h_2(a)\}$ é uma subálgebra de A que contém X. □

5.2 Ordem

Nesta seção descreveremos muito sucintamente alguns conceitos básicos da teoria de ordem, incluíndo as noções de reticulados, álgebra de Heyting e álgebra de Boole.

A teoria de ordem é uma das estruturas fundamentais da matemática moderna, como foi reconhecido pelo célebre grupo francês Bourbaki. Conceitos da teoria de ordem são relevantes em diferentes áreas da matemática; basta mencionar o fundamental Lema de Zorn. Em análise funcional, por exemplo, além do Lema de Zorn, são utilizados conceitos de teoria da ordem para definir o importante conceito de *rede*, que generaliza o conceito de seqüência, permitindo caracterizar as diferentes topologias de um espaço de Hilbert em termos de convergência (em redes). Por outro lado, o estudo do reticulado das projeções numa álgebra de von Neumann mostra outra importante aplicação da teoria de ordem. Conceitos da teoria de ordem (por exemplo, os reticulados contínuos) são também utilizados em Ciências da Computação.

Definição 5.2.1. Seja A um conjunto não vazio. Uma relação binária $\preceq$ sobre A é uma *pré-ordem* se ela é reflexiva e transitiva, isto é:

1. $a \preceq a$ para todo $a \in A$; e

2. $a \preceq b$, $b \preceq c$ implica $a \preceq c$, para todo $a, b, c \in A$.

Exemplo 5.2.2. Defina em Sent (ver Exemplo 5.1.6) a seguinte relação: $\varphi \preceq \psi$ sse $\varphi \models \psi$ (sse $\varphi \Rightarrow \psi$ é uma tautologia). Então $\preceq$ é uma pré-ordem sobre Sent. $\triangle$

Uma pré-ordem não é necesariamente antisimétrica, isto é: $a \preceq b$ e $b \preceq a$ não implica em geral $a = b$. No Exemplo 5.2.2 temos que $(\varphi \wedge \psi) \preceq (\psi \wedge \varphi)$ e $(\psi \wedge \varphi) \preceq (\varphi \wedge \psi)$ mas $(\varphi \wedge \psi) \neq (\psi \wedge \varphi)$.

Esa 'deficiência' nas pré-ordens é evitada exigindo que a pré-ordem seja antisimétrica, chegando-se assim ao conceito de *ordem* (parcial):

Definição 5.2.3. Uma pré-ordem $\preceq$ sobre um conjunto A é uma *ordem (parcial)* se $\preceq$ é antisimétrica, isto é: $a \preceq b$ e $b \preceq a$ implica $a = b$, para todo $a, b \in A$. Nesse caso diremos que $\langle A, \preceq \rangle$ é um *poset*.[1] Em geral, as relações de ordem serão denotadas por $\leq$ no lugar de $\preceq$. Escreveremos $a \npreceq b$ e $a \nleq b$ para indicar que não é o caso que $a \preceq b$ e que não é o caso que $a \leq b$, respectivamente.

A ordem é dita *parcial* porque é possível ter elementos que não são comparáveis, isto é, podem existir elementos $a, b \in A$ tais que $a \nleq b$ e $b \nleq a$.

Exemplo 5.2.4. Seja X um conjunto e $\wp(X)$ o conjunto das partes de X. Então a relação $Y \leq Z$ sse $Y \subseteq Z$ é uma ordem parcial em $\wp(X)$, isto é, $\langle \wp(X), \subseteq \rangle$ é um poset. Se existem $a, b \in X$ tais que $a \neq b$ então $\{a\}$ e $\{b\}$ são dois elementos de $\wp(X)$ que não são comparáveis, logo a ordem $\subseteq$ é parcial. $\triangle$

Se quaisquer dois elementos de um poset são comparáveis, dizemos que a ordem é *total* ou *linear*. Assim, numa ordem linear $\langle A, \leq \rangle$ temos que $a \leq b$ ou $b \leq a$ para todo $a, b \in A$.

Veremos a seguir uma técnica geral para transformar uma pré-ordem numa ordem parcial.

Proposição 5.2.5. *Se $\preceq$ é uma pré-ordem sobre um conjunto A, considere a seguinte relação sobre A: $a \approx b$ sse $a \preceq b$ e $b \preceq a$. Então $\approx$ é uma relação*

[1] Em inglês é um acrônimo para "Partial Ordered Set".

de equivalência em A. Seja $A/_{\approx}$ o quociente de A pela relação $\approx$, isto é, $A/_{\approx} = \{[a] \ : \ a \in A\}$, em que $[a] = \{c \in A \ : \ a \approx c\}$ denota a classe de equivalência de $a \in A$. Considere a seguinte relação em $A/_{\approx}$: $[a] \leq [b]$ sse $a \preceq b$. Então $\langle A/_{\approx}, \leq \rangle$ é um poset.

Demonstração: Deixamos como exercício para o leitor provar que $\approx$ é de fato uma relação de equivalência em A. Provaremos agora que a relação $\leq$ sobre $A/_{\approx}$ está bem definida, isto é, a verdade de '$[a] \leq [b]$' independe dos representantes a e b. Assim, suponha que $a \approx a_1$ e $b \approx b_1$ (isto é, escolhemos $a_1 \in [a]$ e $b_1 \in [b]$ arbitrários). Então $a \preceq b$ implica o seguinte:

$$a_1 \preceq a \preceq b \preceq b_1$$

portanto $a_1 \preceq b_1$. Vemos assim que a relação $\leq$ está bem definida. Provaremos que $\leq$ é uma ordem parcial. A reflexividade e a transitividade de $\leq$ são herdadas de $\preceq$. Suponha que $[a] \leq [b]$ e $[b] \leq [a]$. Então $a \preceq b$ e $b \preceq a$, portanto $a \approx b$, logo $[a] = [b]$. $\square$

Exemplo 5.2.6. Aplicando no Exemplo 5.2.2 a Proposição 5.2.5 obtemos que $\varphi \approx \psi$ sse $\models (\varphi \Rightarrow \psi)$ e $\models (\psi \Rightarrow \varphi)$ sse $\varphi \equiv \psi$, pelo Exercício 2.2.6(2). Logo $\langle \mathsf{Sent}/_{\approx}, \leq \rangle$ é um poset, em que $[\varphi] \leq [\psi]$ sse $\varphi \models \psi$, com

$$[\varphi] = \{\gamma \ : \ \varphi \approx \gamma\} = \{\gamma \ : \ \varphi \equiv \gamma\}.$$

$\triangle$

Estabeleceremos agora um princípio fundamental na teoria de ordens: o *princípio de dualidade*. Para isso introduzimos o seguinte conceito.

Definição 5.2.7. Seja $\mathtt{P} = \langle A, \leq \rangle$ um poset. O *poset dual de* $\mathtt{P}$ é o poset $\mathtt{P}^{op} = \langle A, \leq^{op} \rangle$ tal que $a \leq^{op} b$ sse $b \leq a$, para todo $a, b \in A$.

Claramente $\mathtt{P}^{op}$ é um poset. Definimos o seguinte: $a \geq b$ sse $b \leq a$. Logo, $a \leq^{op} b$ sse $a \geq b$.

Dada uma afirmação Φ sobre um poset $\mathtt{P} = \langle A, \leq \rangle$, obtemos a *afirmação dual* Φ^{op}, que é a afirmação obtida de Φ substituíndo (simultaneamente) cada ocorrência de $\leq$ por $\geq$ e cada ocorrência de $\geq$ por $\leq$. Observe que $\Phi = (\Phi^{op})^{op}$, e

$$\Phi \text{ vale em } \mathtt{P} \text{ sse } \Phi^{op} \text{ vale em } \mathtt{P}^{op} \qquad (*)$$

Obtemos então o seguinte resultado fundamental da teoria de ordens:

Teorema 5.2.8 (Princípio de Dualidade)**.** *Seja Φ uma propriedade de posets. Se Φ vale para todo poset* $\mathtt{P}$ *então Φ^{op} vale para todo poset* $\mathtt{P}$*.*

Demonstração: Seja Φ uma propriedade válida em todo poset $\mathtt{P}$, e fixe um poset $\mathtt{P}$. Então, por $(*)$ temos que Φ^{op} vale em $\mathtt{P}$ sse $(\Phi^{op})^{op}$ vale em $\mathtt{P}^{op}$ sse Φ vale em $\mathtt{P}^{op}$. Isto é,

$$\Phi^{op} \text{ vale em } \mathtt{P} \text{ sse } \Phi \text{ vale em } \mathtt{P}^{op} \qquad (**)$$

Dado que $\mathtt{P}^{op}$ é um poset e Φ vale em todo poset (por hipótese), então inferimos que Φ^{op} vale em $\mathtt{P}$, por $(**)$. $\square$

O significado da propriedade $(*)$ e do Princípio de Dualidade é o seguinte: cada definição D estabelecida num poset determina imediatamente uma outra afirmação, a afirmação dual D^{op}. E cada propriedade Φ demonstravelmente válida em todo poset determina imediatamente outra propriedade demonstravelmente válida em todo poset, a propriedade dual Φ^{op}. Assim, as definições e os resultados demonstrados em posets vêm 'de a pares': cada definição ou resultado sobre posets determina imediatamente sua definição ou resultado dual sobre posets. Este fato será utilizado constantemente, sendo uma característica básica da teoria de ordens.

Introduziremos agora alguns conceitos fundamentais da teoria de ordens, com os consequentes conceitos duais:

Definição 5.2.9. Seja $\langle A, \leq \rangle$ um poset e $X \subseteq A$ um subconjunto.
(1) Um elemento $a \in A$ é dito um *limitante superior de* X se

$$x \leq a \quad \text{para todo } x \in X.$$

O conjunto dos limitantes superiores de X é denotado por $Lsup(X)$. Note que $Lsup(\emptyset) = A$.
(2) Um elemento $a \in A$ é dito *máximo de* X se $a \in (X \cap Lsup(X))$.

Note que o máximo de um conjunto X, se existir, é único:

Proposição 5.2.10. *Para todo $X \subseteq A$ temos que $X \cap Lsup(X)$ tem, no máximo, um elemento.*

Demonstração: Suponha que $(X \cap Lsup(X)) \neq \emptyset$, e sejam $a, b \in (X \cap Lsup(X))$. Dado que $a \in X$ e $b \in Lsup(X)$ então $a \leq b$. Por um argumento similar (trocando os papéis de a e b) vemos que $b \leq a$. Daqui $a = b$. $\square$

Se existir o máximo do conjunto X, então o denotaremos por $Max^{\mathtt{P}}(X)$ ou simplesmente $Max(X)$ (quando `P` for óbvio no contexto). Se existir o máximo de A, este será denotado por $\top$.

Note que nem todo subconjunto num poset tem máximo: por exemplo o intervalo $[0,1) = \{x \in \mathbb{R} \,:\, 0 \leq x < 1\}$ é um subconjunto de $\mathbb{R}$ que não tem máximo (com relação à ordem usual dos números reais). Evidentemente se $Lsup(X) = \emptyset$ então X não tem máximo. Em particular, o conjunto vazio $\emptyset$ não tem máximo.

Podemos dualizar as definições e resultados anteriores, obtendo o seguinte:

Definição 5.2.11. Seja $\langle A, \leq \rangle$ um poset e $X \subseteq A$ um subconjunto.
(1) Um elemento $a \in A$ é dito um *limitante inferior de* X se

$$a \leq x \;\text{ para todo } x \in X.$$

O conjunto dos limitantes inferiores de X é denotado por $Linf(X)$. Note que $Linf(\emptyset) = A$.
(2) Um elemento $a \in A$ é dito *mínimo de* X se $a \in (X \cap Linf(X))$.

Proposição 5.2.12. *Para todo $X \subseteq A$ temos que $X \cap Linf(X)$ tem, no máximo, um elemento.*

Demonstração: Conseqüência do Princípio de Dualidade. □

Se existir o mínimo do conjunto X, então o denotaremos por $Min^{\mathtt{P}}(X)$ ou simplesmente $Min(X)$. Se existir o mínimo de A, este será denotado por $\perp$.

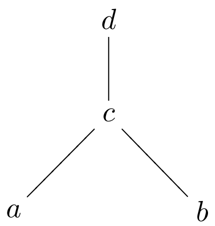
Às vezes requerir a existência de máximo de um conjunto é uma exigência muito forte. No exemplo do intervalo $[0,1)$, claramente 1 é o limitante superior 'mais próximo' do intervalo, e deveria ser considerado um tipo de 'máximo idealizado' do conjunto. A noção de 'máximo idealizado' de um conjunto é capturada pela seguinte definição:

Definição 5.2.13. Seja $\mathtt{P} = \langle A, \leq \rangle$ um poset e $X \subseteq A$ um subconjunto. O *supremo de X em $\mathtt{P}$*, se existir, é definido como sendo o elemento $\bigvee^{\mathtt{P}} X$ de A dado por $Min(Lsup(X))$.

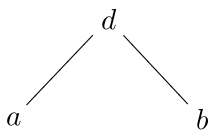
Quando `P` for óbvio no contexto escreveremos $\bigvee X$ no lugar de $\bigvee^{\mathtt{P}} X$. Note que, se X tem máximo, então X tem supremo em `P`, e $Max(X) = \bigvee X$.

Observação 5.2.14. O uso do supraíndice `P` em $\bigvee^{\mathtt{P}}$ é fundamental quando tem mais de um poset sendo considerado. Se $\mathtt{P} = \langle A, \leq \rangle$ e $\mathtt{P}' = \langle A', \leq \rangle$ tal que $A' \subseteq A$ (e a ordem de $\mathtt{P}$ estende a ordem de $\mathtt{P}'$) então $\bigvee^{\mathtt{P}'} X \neq \bigvee^{\mathtt{P}} X$ em geral. Por exemplo, considere os posets $\mathtt{P} = \langle \mathbb{R}, \leq \rangle$ e $\mathtt{P}' = \langle \mathbb{Q}, \leq \rangle$ dos números reais e os números racionais com a sua ordem usual. Seja $X = \{x \in \mathbb{Q} \; : \; x^2 \leq 2\}$. Então $\bigvee^{\mathtt{P}} X = \sqrt{2}$ mas não existe $\bigvee^{\mathtt{P}'} X$.

Exemplo 5.2.15. Outro exemplo em que o supremo depende do poset em que é calculado é o seguinte: considere um poset $\mathtt{P}$ contendo quatro elementos ordenados da maneira seguinte (no diagrama a seguir, um elemento x encontra-se debaixo de um elemento y sse $x \leq y$):



Seja $A' = \{a, b, d\}$ e $\mathtt{P}' = \langle A', \leq \rangle$ com a ordem herdada de $\mathtt{P}$.



Considere agora $X = \{a, b\}$. Então $\bigvee^{\mathtt{P}} X = c$ mas $\bigvee^{\mathtt{P}'} X = d$. △

Por dualidade, definimos o 'mínimo idealizado' de um conjunto X através do conceito de ínfimo.

Definição 5.2.16. Seja $\mathtt{P} = \langle A, \leq \rangle$ um poset e $X \subseteq A$ um subconjunto. O *ínfimo de X em $\mathtt{P}$*, se existir, é definido como sendo o elemento $\bigwedge^{\mathtt{P}} X$ de A dado por $Max(Linf(X))$.

Como antes, quando $\mathtt{P}$ for óbvio escreveremos $\bigwedge X$ no lugar de $\bigwedge^{\mathtt{P}} X$. Se X tem mínimo, então X tem ínfimo em $\mathtt{P}$, e $Min(X) = \bigwedge X$.

Uma observação interessante é a seguinte: se existir $\bigvee \emptyset$ então $\bigvee \emptyset = \bot$. Por dualidade, se existir $\bigwedge \emptyset$ então $\bigwedge \emptyset = \top$. Por outro lado, se existir $\bigvee A$ ($\bigwedge A$, respectivamente) então $\bigvee A = \top$ ($\bigwedge A = \bot$, respectivamente).

Claramente podemos caracterizar supremos e ínfimos da seguinte maneira:

Proposição 5.2.17. *Seja $\mathtt{P} = \langle A, \leq \rangle$ um poset e $X \subseteq A$ um subconjunto.*
(i) $a \in A$ é o supremo de X em $\mathtt{P}$ sse satisfaz o seguinte:

1. $x \leq a$ para todo $x \in X$;

2. dado $b \in A$, se $x \leq b$ para todo $x \in X$ então $a \leq b$.

(ii) $a \in A$ é o ínfimo de X em $\mathtt{P}$ sse satisfaz o seguinte:

1. $a \leq x$ para todo $x \in X$;

2. dado $b \in A$, se $b \leq x$ para todo $x \in X$ então $b \leq a$.

5.3 Reticulados

Um caso particularmente interesante de poset são os reticulados.

Definição 5.3.1. Seja $\mathtt{L} = \langle A, \leq \rangle$ um poset. Se existem $\bigvee\{x, y\}$ e $\bigwedge\{x, y\}$ em $\mathtt{L}$ para todo $x, y \in A$ então dizemos que $\mathtt{L}$ é um *reticulado*.

Exemplo 5.3.2. Considere o poset $\langle \mathsf{Sent}/_{\approx}, \leq \rangle$ do Exemplo 5.2.6. Dados $[\varphi], [\psi] \in \mathsf{Sent}/_{\approx}$ considere o elemento $[\varphi \vee \psi] \in \mathsf{Sent}/_{\approx}$. Dado que $\varphi \models \varphi \vee \psi$ e $\psi \models \varphi \vee \psi$ então $[\varphi] \leq [\varphi \vee \psi]$ e $[\psi] \leq [\varphi \vee \psi]$. Por outro lado, seja $[\gamma] \in \mathsf{Sent}/_{\approx}$ tal que $[\varphi] \leq [\gamma]$ e $[\psi] \leq [\gamma]$. Então $\varphi \models \gamma$ e $\psi \models \gamma$, portanto $\varphi \vee \psi \models \gamma$. [2] Daqui inferimos que $[\varphi \vee \psi] \leq [\gamma]$, portanto $[\varphi \vee \psi] = \bigvee\{[\varphi], [\psi]\}$, pela Proposição 5.2.17(i).

Analogamente podemos provar que $[\varphi \wedge \psi] = \bigwedge\{[\varphi], [\psi]\}$ (deixamos isto como exercício para o leitor), portanto $\langle \mathsf{Sent}/_{\approx}, \leq \rangle$ é um reticulado. $\triangle$

Observação 5.3.3. Um reticulado $\mathtt{L}$ pode ser visto (e frequentemente isto resulta muito útil) como uma álgebra $\mathtt{L} = \langle A, \sqcup, \sqcap \rangle$ em que $\sqcup$ e $\sqcap$ são duas operações binárias. Nesse caso escreveremos $x \sqcup y$ e $x \sqcap y$ para denotar $\bigvee\{x, y\}$ e $\bigwedge\{x, y\}$, respectivamente. Note que $x \sqcup y = y \sqcup x$, e $x \sqcap y = y \sqcap x$ para todo $x, y \in A$. Mais ainda, $x \leq y$ sse $x \sqcup y = y$ sse $x \sqcap y = x$.

Os seguintes resultados justificam que possamos considerar reticulados como álgebras.

Proposição 5.3.4. *Seja $\mathtt{L} = \langle A, \leq \rangle$ um reticulado, e considere as operações $\sqcup, \sqcap : A \times A \to A$ definidas na Observação 5.3.3. Então $\langle A, \sqcup, \sqcap \rangle$ é uma álgebra satisfazendo o seguinte, para todo $a, b, c \in A$:*

[2] Para provar isto podemos usar argumentos semânticos ou simplesmente utilizar argumentos sintácticos, digamos derivações em PC, e usar o teorema de adequação.

$[L1]$	$(a \sqcup b) \sqcup c = a \sqcup (b \sqcup c)$	$[L1]^{op}$	$(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c)$
$[L2]$	$a \sqcup b = b \sqcup a$	$[L2]^{op}$	$a \sqcap b = b \sqcap a$
$[L3]$	$a \sqcup a = a$	$[L3]^{op}$	$a \sqcap a = a$
$[L4]$	$a \sqcap (a \sqcup b) = a$	$[L4]^{op}$	$a \sqcup (a \sqcap b) = a$

Proposição 5.3.5. *Seja* $\mathtt{L} = \langle A, \sqcup, \sqcap \rangle$ *uma álgebra em que* $\sqcup$ *e* $\sqcap$ *são duas operações binárias satisfazendo as propriedades da Proposição 5.3.4.*
(i) Para todo $a, b \in A$ *temos que* $a \sqcup b = b$ *sse* $a \sqcap b = a$.
(ii) Defina uma relação binária $\leq$ *em* A *como segue:* $a \leq b$ *sse* $a \sqcap b = a$. *Então* $\langle A, \leq \rangle$ *é um reticulado tal que* $\bigvee\{a, b\} = a \sqcup b$ *e* $\bigwedge\{a, b\} = a \sqcap b$.

A partir das proposições anteriores (cuja demonstração deixamos como exercício para o leitor) vemos que um reticulado pode ser apresentado em termos de teoria da ordem ou em termos de álgebras.

A partir de agora consideraremos os reticulados como álgebras com duas operações binárias. Mais ainda, se o reticulado possui máximo $\top$ ou mínimo $\bot$ então eses elementos serão considerados como operações 0-árias.

Observação 5.3.6. Considerando reticulados como álgebras, podemos formular o seguinte *princípio de dualidade* para reticulados: dada uma afirmação Φ sobre um reticulado, a *afirmação dual* Φ^{op} é a afirmação obtida de Φ substituindo (simultaneamente) cada ocorrência de $\leq, \geq, \sqcup, \sqcap, \top$ e $\bot$ por $\geq, \leq, \sqcap, \sqcup, \bot$ e $\top$, respectivamente. Então vale o seguinte:

Se Φ *vale para todo reticulado então* Φ^{op} *vale para todo reticulado.*

Uma propriedade desejável em reticulados é a distributividade de supremos por ínfimos, e vice-versa. Temos o seguinte (as provas são deixadas como exercício para o leitor):

Proposição 5.3.7. *Seja* $\mathtt{L} = \langle A, \sqcup, \sqcap \rangle$ *um reticulado. Então vale o seguinte, para todo* $a, b, c \in A$:

$$a \sqcup (b \sqcap c) \leq (a \sqcup b) \sqcap (a \sqcup c)$$

Por dualidade, obtemos o seguinte:

Corolário 5.3.8. *Seja* $\mathtt{L} = \langle A, \sqcup, \sqcap \rangle$ *um reticulado. Então vale o seguinte, para todo* $a, b, c \in A$:

$$a \sqcap (b \sqcup c) \geq (a \sqcap b) \sqcup (a \sqcap c)$$

Proposição 5.3.9. *Num reticulado* $\mathtt{L} = \langle A, \sqcup, \sqcap \rangle$ *são equivalentes:*

$$\begin{array}{lll} [D] & a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c) & \textit{para todo } a, b, c \in A \\ [D]^{op} & a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c) & \textit{para todo } a, b, c \in A \end{array}$$

Note que não podemos aplicar o princípio de dualidade para provar a proposição acima.

Definição 5.3.10. Dizemos que um reticulado $\mathtt{L} = \langle A, \sqcup, \sqcap \rangle$ é *distributivo* se satisfaz a propriedade $[D]$ (ou equivalentemente, a propriedade $[D]^{op}$) da Proposição 5.3.9.

Exemplo 5.3.11. Considere o reticulado $\langle \mathsf{Sent}/_{\approx}, \leq \rangle$ do Exemplo 5.3.2. Pelo Exemplo 4.3.3 e pela completude do método de sequentes temos que $\langle \mathsf{Sent}/_{\approx}, \leq \rangle$ é um reticulado distributivo. Mais ainda, o reticulado tem mínimo dado por $[\,\mathbf{f}\,]$. $\triangle$

Analisaremos agora o importante conceito de pseudo-complemento em reticulados, que está relacionado com o conceito de negação e de implicação em lógicas, como veremos ainda nesta seção.

Definição 5.3.12. Seja $\mathtt{L} = \langle A, \sqcup, \sqcap, \bot \rangle$ um reticulado com mínimo $\bot$. O $\sqcap$-complemento de $a \in A$ (se existe) é o elemento $-a$ de A dado por $Max(\{c \in A \ : \ c \sqcap a = \bot\})$.

Proposição 5.3.13. *O* $\sqcap$*-complemento* $-a$ *de* a*, se existe, é único.*

Por dualidade, obtemos a seguinte

Definição 5.3.14. Seja $\mathtt{L} = \langle A, \sqcup, \sqcap, \top \rangle$ um reticulado com máximo $\top$. O $\sqcup$-complemento de $a \in A$ (se existe) é o elemento $-a$ de A dado por $Min(\{c \in A \ : \ c \sqcup a = \top\})$.

Proposição 5.3.15. *O* $\sqcup$*-complemento* $-a$ *de* a*, se existe, é único.*

Definição 5.3.16. Um elemento a num reticulado $\mathtt{L} = \langle A, \sqcup, \sqcap, \bot, \top \rangle$ com máximo e mínimo é dito *complementado* se tem um elemento b que é simultaneamente $\sqcap$-complemento e $\sqcup$-complemento de a. O reticulado $\mathtt{L}$ é dito *complementado* se todo elemento é complementado.

Observação 5.3.17. Se $\mathtt{L}$ é complementado então existe uma operação $- : A \to A$ tal que, para todo $a \in A$:

$$-a \sqcap a = \bot \qquad\qquad -a \sqcup a = \top$$

Num reticulado arbitrário é possível que um elemento a tenha um $\sqcap$-complemento b e um $\sqcup$-complemento c tais que $b \neq c$.

Nos reticulados distributivos a propriedade do complemento mencionada na Observação 5.3.17 caracteriza o complemento:

Proposição 5.3.18. *Se* $\mathtt{L}$ *é distributivo e possui uma operação* $-: A \to A$ *tal que, para todo* $a \in A$*:*

$$-a \sqcap a = \bot \quad e \quad -a \sqcup a = \top$$

então $-a$ *é o complemento de* a*, portanto* $\mathtt{L}$ *é complementado.*

A noção de $\sqcap$-complemento pode ser generalizada da seguinte maneira:

Definição 5.3.19. Seja $\mathtt{L} = \langle A, \sqcup, \sqcap \rangle$ um reticulado, e sejam $a, b \in A$. O *pseudo-complemento de* a *relativo a* b, denotado por $a \sqsupset b$, é o elemento de A (se existe) dado por $Max(\{c \in A \ : \ c \sqcap a \leq b\})$. Se existe o pseudo complemento $a \sqsupset b$ para todo $a, b \in A$ dizemos que $\mathtt{L}$ é *relativamente pseudo-complementado.*

Observe que $a \sqsupset b$ satisfaz, para todo $c \in A$:

$$c \leq a \sqsupset b \ \text{ sse } \ c \sqcap a \leq b.$$

Da definição anterior é fácil provar que, se $\mathtt{L}$ é relativamente pseudo-complementado então $\mathtt{L}$ tem máximo $\top$, dado por $\top = a \sqsupset a$ para qualquer $a \in A$. Mais ainda, pode-se provar o seguinte:

Proposição 5.3.20. *Todo reticulado relativamente pseudo-complementado é distributivo.*

Chegamos agora a uma importante definição:

Definição 5.3.21. Uma *álgebra de Heyting* é um reticulado relativamente pseudo-complementado com elemento mínimo.

A partir das definições vemos que numa álgebra de Heyting existe o $\sqcap$-complemento de a, dado por $-a := a \sqsupset \bot$, e satisfazendo $-a \sqcap a = \bot$ para todo $a \in A$. Porém, não necessariamente vale que $-a \sqcup a = \top$. Desta maneira, uma álgebra de Heyting é uma álgebra $\mathcal{H} = \langle A, \sqcup, \sqcap, \sqsupset, \bot \rangle$ (note que $\top$ e $-$ são operações *definidas* a partir das outras).

Podemos caracterizar uma álgebra de Heyting equacionalmente, isto é, através de equações que satisfazem as suas operações (lembrando que a relação $\leq$ também é dada a partir de uma equação):

Proposição 5.3.22. *Seja* $\mathcal{H} = \langle A, \sqcup, \sqcap, \sqsupset, \bot \rangle$ *uma álgebra abstrata. Então* $\mathcal{H}$ *é uma álgebra de Heyting sse* $\mathcal{H}$ *satisfaz o seguinte:*

1. $\mathcal{H}$ é um reticulado distributivo com mínimo $\perp$, isto é, $a \sqcap \perp = \perp$ para todo $a \in A$ (sendo que $a \leq b$ sse $a \sqcap b = a$ para todo $a, b \in A$);

2. $a \sqcap (a \sqsupset b) \leq b$ para todo $a, b \in A$;

3. $a \sqcap (b \sqsupset c) = a \sqcap ((a \sqcap b) \sqsupset (a \sqcap c))$ para todo $a, b, c \in A$;

4. $a \leq (b \sqcap c) \sqsupset c$ para todo $a, b, c \in A$.

Exemplo 5.3.23. Considere o reticulado $\langle \mathsf{Sent}/_{\approx}, \leq \rangle$ do Exemplo 5.3.11. Já foi provado nesse exemplo que este reticulado é distributivo e tem mínimo $\perp$. Considere a operação binária $\sqsupset$: $\mathsf{Sent}/_{\approx} \times \mathsf{Sent}/_{\approx} \to \mathsf{Sent}/_{\approx}$ dada por: $[\varphi] \sqsupset [\psi] := [\varphi \Rightarrow \psi]$ para todo $[\varphi], [\psi] \in \mathsf{Sent}/_{\approx}$. É fácil provar que esta operação está bem definida (exercício para o leitor). Por outro lado, usando o Teorema da Dedução temos que, para todo $[\varphi], [\psi], [\gamma] \in \mathsf{Sent}/_{\approx}$,

$$[\gamma] \sqcap [\varphi] \leq [\psi] \text{ sse } [\gamma] \leq [\varphi] \sqsupset [\psi]$$

pois $\gamma \wedge \varphi \models \psi$ sse $\gamma \models \varphi \Rightarrow \psi$. Assim, temos que $\langle \mathsf{Sent}/_{\approx}, \leq \rangle$ é uma álgebra de Heyting, podendo ser escrita como uma álgebra $\langle \mathsf{Sent}/_{\approx}, \sqcup, \sqcap, \sqsupset, \perp \rangle$ tal que

$$[\varphi] \sqcup [\psi] = [\varphi \vee \psi] \qquad [\varphi] \sqcap [\psi] = [\varphi \wedge \psi]$$

$$[\varphi] \sqsupset [\psi] = [\varphi \Rightarrow \psi] \qquad \perp = [\mathbf{f}].$$

Lembre que as operações $-[\varphi] := [\varphi] \sqsupset \perp$ e $\top := [p_0 \Rightarrow p_0]$ são definidas, e não são primitivas. $\triangle$

Estabelecemos a seguir algumas propriedades das álgebras de Heyting, cuja prova deixamos para o leitor:

Proposição 5.3.24. *Seja $\mathcal{H} = \langle A, \sqcup, \sqcap, \sqsupset, \perp \rangle$ uma álgebra de Heyting. Então vale o seguinte, para todo $a, b \in A$ (lembrando que $-a := a \sqsupset \perp$ e $\top := a \sqsupset a$):*
(1) $a \leq b$ implica $-b \leq -a$, mas não vale a recíproca.
(2) $-\perp = \top$; $-\top = \perp$.
(3) $-a \sqcap a = \perp$.
(4) $-(-a \sqcap a) = \top$ mas $-a \sqcup a \neq \top$ em geral.
(5) $a \leq - - a$, mas não vale a recíproca.

(6) $-a = ---a$.
(7) $a \sqsupset b \leq -b \sqsupset -a$*, mas não vale a recíproca.*

Veremos na próxima seção que todas estas propriedades correspondem com leis da chamada *lógica intuicionista*.

Definimos finalmente uma noção de importância central para a algebrização da lógica proposicional clássica:

Definição 5.3.25. Um reticulado com $\sqcap$-complemento e máximo $\top$ é uma *álgebra de Boole* se o $\sqcap$-complemento é também um $\sqcup$-complemento. Equivalentemente, uma álgebra de Boole é um reticulado distributivo complementado (portanto tem $\top$ e $\bot$).

Uma álgebra de Boole é uma álgebra de Heyting satisfazendo uma propriedade equacional adicional, como prova o seguinte resultado:

Proposição 5.3.26. *Seja* $\mathcal{H} = \langle A, \sqcup, \sqcap, \sqsupset, \bot \rangle$ *uma álgebra de Heyting. São equivalentes:*
(1) $\mathcal{H}$ *é uma álgebra de Boole.*
(2) $-a \sqcup a = \top$ *para todo* $a \in A$.
(3) $--a = a$ *para todo* $a \in A$.
(4) $a \sqsupset b = -b \sqsupset -a$ *para todo* $a, b \in A$.
(5) $a \sqsupset b = -a \sqcup b$ *para todo* $a, b \in A$.

5.4 Semântica algébrica

Finalmente, aplicaremos todos os resultados algébricos obtidos até agora para dar uma nova caracterização semântica da LPC.

Proposição 5.4.1. *A álgebra de Heyting* $\langle \mathsf{Sent}/_{\approx}, \sqcup, \sqcap, \sqsupset, \bot \rangle$ *do Exemplo 5.3.23 é, de fato, uma álgebra de Boole.*

A prova da proposição anterior é deixada como exercício para o leitor.

Definição 5.4.2. A álgebra de Boole $\langle \mathsf{Sent}/_{\approx}, \sqcup, \sqcap, \sqsupset, \bot \rangle$ é chamada de *álgebra de Lindembaum-Tarski* da LPC, em honor aos seus descobridores. Alternativamente podemos apresentar esta álgebra sobre a assinatura C^7 tal que $|C^7| = \{\vee, \wedge, \neg, \mathbf{t}, \mathbf{f}\}$, obtendo a álgebra de Boole

$$\mathcal{B}_{\mathsf{LPC}} := \langle \mathsf{Sent}_{+}/_{\approx}, \sqcup, \sqcap, -, \top, \bot \rangle$$

em que $\mathsf{Sent}_{+} := L(C^7)$. A álgebra $\mathcal{B}_{\mathsf{LPC}}$ também será chamada de álgebra de Lindembaum-Tarski da LPC.

Veremos a seguir que a LPC é correta e completa com relação à semântica de valorações em álgebras de Boole, sendo que a álgebra de Lindembaum-Tarski junto com a projeção canônica $\varphi \mapsto [\varphi]$ constitui o modelo canônico que fornece uma outra prova da completude do método axiomático com relação à semântica clássica.

Definição 5.4.3. Seja $\mathcal{B} = \langle |\mathcal{B}|, \sqcup, \sqcap, -, \top, \bot \rangle$ uma álgebra de Boole (sobre a assinatura C^7). Definimos uma *valoração sobre* $\mathcal{B}$ como sendo um homomorfismo de álgebras $v : \mathsf{Sent}_+ \to |\mathcal{B}|$.

Note que uma valoração v sobre $\mathcal{B}$ é caracterizada apenas por uma função $v_0 : \mathcal{V} \to |\mathcal{B}|$. De fato, v é a única extensão de v_0 para um homomorfismo com domínio Sent_+. Escreveremos $[\![\varphi]\!]_v$ no lugar de $v(\varphi)$.

Definição 5.4.4. Seja φ uma sentença em Sent_+.
(1) Seja $\mathcal{B}$ uma álgebra de Boole. Dizemos que φ é *válida em* $\mathcal{B}$ se $[\![\varphi]\!]_v = \top$ para toda valoração v sobre $\mathcal{B}$.
(2) Dizemos que φ é *válida* (com relação à semântica em álgebras de Boole) se φ é válida em toda álgebra de Boole.

Exemplo 5.4.5. Seja $\mathcal{B}_{\mathsf{LPC}}$ a álgebra de Lindembaum-Tarski da LPC. Considere a projeção canônica $v_{can} : \mathsf{Sent}_+ \to \mathsf{Sent}_+/_{\approx}$ dada por $v_{can}(\varphi) = [\varphi]$. Então v_{can} é um homomorphismo de álgebras, isto é, v_{can} é uma valoração sobre a álgebra de Boole $\mathcal{B}_{\mathsf{LPC}}$. $\triangle$

Finalmente, chegamos ao principal resultado sobre semântica algébrica da LPC e a sua conexão com a álgebra de Lindembaum-Tarski:

Teorema 5.4.6. *Seja φ uma sentença em Sent_+. São equivalentes:*
(1) φ é válida (com relação à semântica em álgebras de Boole).
(2) $v_{can}(\varphi) = \top$.
(3) φ é tautologia (isto é, φ é válida em $\mathcal{B}_2$).
(4) φ é teorema de PC.

A prova deste resultado pode ser achada em, por exemplo, [12].

Observe que (2)$\to$(4) ou, equivalentemente, $\neg$(4)$\to$ $\neg$(2), é a chave para a prova do teorema de completude (fraca) de PC com relação à semântica em álgebras de Boole. Com efeito, assumindo (2)$\to$(4) temos o seguinte: se $\nvdash_{\mathsf{PC}} \varphi$ então $v_{can}(\varphi) \neq \top$, logo existe um modelo em álgebras de Boole que não satisfaz φ.

Podemos estender o resultado anterior para a completude forte (assumindo a compacidade da LPC).

Definição 5.4.7. Seja $\Gamma \cup \{\varphi\}$ um subconjunto finito de Sent_+.
(1) Seja $\mathcal{B}$ uma álgebra de Boole. Dizemos que φ *é consequência semântica de* Γ *em* $\mathcal{B}$, e o denotamos por $\Gamma \models_{\mathcal{B}} \varphi$, se para toda valoração v sobre $\mathcal{B}$ temos que $\bigwedge\{[\![\gamma]\!]_v \ : \ \gamma \in \Gamma\} \leq [\![\varphi]\!]_v$.
(2) Dizemos que φ *é consequência semântica de* Γ (com relação à semântica em álgebras de Boole), e o denotamos por $\Gamma \models_{AB} \varphi$, se $\Gamma \models_{\mathcal{B}} \varphi$ para toda álgebra de Boole $\mathcal{B}$.

Note que $\Gamma \models \varphi$ sse $\Gamma \models_{\mathcal{B}_\mathbf{2}} \varphi$, com $\mathcal{B}_\mathbf{2}$ a álgebra de Boole com domínio $\mathbf{2}$. Logo, na Definição 5.4.7(1) estamos generalizando a noção de conseqüência semântica, de $\mathcal{B}_\mathbf{2}$ para álgebras de Boole em geral. Obtemos como conseqüência imediata do Teorema 5.4.6 o seguinte:

Corolário 5.4.8. *Seja* $\Gamma \cup \{\varphi\}$ *um subconjunto finito de* Sent_+. *São equivalentes:*
(1) $\Gamma \models_{AB} \varphi$.
(2) $\bigwedge\{v_{can}(\gamma) \ : \ \gamma \in \Gamma\} \leq v_{can}(\varphi)$.
(3) $\Gamma \models_{\mathcal{B}_\mathbf{2}} \varphi$.
(4) $\Gamma \vdash_{\mathsf{PC}} \varphi$.

Desta maneira, vemos que testar conseqüência semântica em *todas* as álgebras de Boole equivale a testar conseqüência semântica com relação à valoração canônica na álgebra de Lindembaum-Tarski, que por sua vez equivale a testar conseqüência semântica em $\mathcal{B}_\mathbf{2}$. Assim, $\mathcal{B}_\mathbf{2}$ é um tipo de álgebra de Boole canônica. O mesmo resultado *não* vale se consideramos álgebras de Heyting e a sua lógica associada, a lógica intuicionista.

A lógica proposicional intuicionista (LPI) foi introduzida por Heyting como uma formalização das idéias filosóficas de Brouwer sobre o intuicionismo e o construtivismo em matemáticas. A LPI é uma lógica estritamente contida na LPC, no sentido que toda inferência realizada na LPI pode ser realizada na LPC, mas a recíproca não é sempre verdadeira. Por exemplo, um princípio clássico que não vale na LPI é o princípio do terceiro excluído: $\neg\varphi \vee \varphi$ ou, equivalentemente, a lei de dupla negação: $\neg\neg\varphi \Rightarrow \varphi$. Introduzindo a notação $\neg\varphi$ para denotar a fórmula $(\varphi \Rightarrow \mathbf{f})$, podemos axiomatizar a LPI na assinatura C^5, lembrando que $|C^5| = \{\vee, \wedge, \Rightarrow, \mathbf{f}\}$ e $L(C^5) = \mathsf{Sent}$, da maneira seguinte:

(Axio1) $\varphi \Rightarrow (\psi \Rightarrow \varphi)$

(Axio2) $(\varphi \Rightarrow (\psi \Rightarrow \gamma)) \Rightarrow ((\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \gamma))$

(Axio3) $\varphi \Rightarrow (\psi \Rightarrow (\varphi \wedge \psi))$

(Axio4) $(\varphi \wedge \psi) \Rightarrow \varphi$

(Axio5) $(\varphi \wedge \psi) \Rightarrow \psi$

(Axio6) $\varphi \Rightarrow (\varphi \vee \psi)$

(Axio7) $\psi \Rightarrow (\varphi \vee \psi)$

(Axio8) $(\varphi \Rightarrow \gamma) \Rightarrow ((\psi \Rightarrow \gamma) \Rightarrow ((\varphi \vee \psi) \Rightarrow \gamma))$

(Axio9) $(\psi \Rightarrow \neg\varphi) \Rightarrow ((\psi \Rightarrow \varphi) \Rightarrow \neg\psi)$

(Axio10) $\varphi \Rightarrow (\neg\varphi \Rightarrow \psi)$

(MP) $\dfrac{\varphi \qquad \varphi \Rightarrow \psi}{\psi}$

Lembre que $\neg\varphi$ denota a fórmula $(\varphi \Rightarrow \mathbf{f})$. Denominaremos de PI o sistema axiomático acima definido, denotando por $\Gamma \vdash_{\mathsf{PI}} \varphi$ a existência de uma derivação em PI da fórmula φ a partir do conjunto de premissas Γ. O subsistema de PI obtido eliminando o axioma (Axio10) determina a chamada *Lógica minimal de Johánsson* (cf. [9]). Esta última trata-se de uma lógica ainda mais restrita que a lógica intuicionista, que apresenta caráter *paraconsistente* (cf. [4]), no sentido seguinte: de uma contradição $\{\varphi, \neg\varphi\}$ não derivamos *toda* fórmula ψ, mas apenas as fórmulas da forma $\neg\gamma$.

É importante salientar que tanto PI quanto a lógica minimal de Johánsson satisfazem o Teorema da Dedução. De fato, vale o seguinte resultado geral sobre sistemas axiomáticos, cuja prova pode ser encontrada em [10]:

Proposição 5.4.9. *Seja S um sistema axiomático sobre uma assinatura contendo um conectivo binário $\Rightarrow$ tal que os axiomas (Axio1) e (Axio2) são demonstráveis em S, e tal que (MP) é a única regra de inferência de S. Então S satisfaz o Teorema da Dedução.*

Resultados análogos aos do Teorema 5.4.6 e do Corolário 5.4.8 podem ser obtidos: assim, podemos definir no conjunto Sent a relação de equivalência $\varphi \approx_I \psi$ sse $\vdash_{\mathsf{PI}} \varphi \Rightarrow \psi$ e $\vdash_{\mathsf{PI}} \psi \Rightarrow \varphi$, provando que

$$\mathcal{H}_{\mathsf{LPI}} := \langle \mathsf{Sent}/_{\approx_I}, \sqcup, \sqcap, \sqsupset, \bot \rangle$$

é uma álgebra de Heyting tal que $[\varphi] \leq [\psi]$ sse $\varphi \vdash_{\mathsf{PI}} \psi$, e

$$[\varphi] \sqcup [\psi] = [\varphi \vee \psi] \qquad [\varphi] \sqcap [\psi] = [\varphi \wedge \psi]$$

$$[\varphi] \sqsupset [\psi] = [\varphi \Rightarrow \psi] \qquad \bot = [\mathbf{f}]$$

lembrando que $-[\varphi] := [\varphi] \sqsupset \bot$ (logo $-[\varphi] = [\neg\varphi]$) e $\top = [p_0 \Rightarrow p_0]$. A álgebra de Heyting $\mathcal{H}_{\mathsf{LPI}}$ é chamada de *álgebra de Lindembaum-Tarski* da LPI. A seguir é fácil adaptar as Definições 5.4.3 e 5.4.4 obtendo os conceitos de *valoração* sobre uma álgebra de Heyting $\mathcal{H}$, de valoração canônica v^I_{can}, de fórmula *válida* (numa álgebra de Heyting $\mathcal{H}$) e de fórmula *válida* (com relação à semântica em álgebras de Heyting). Assim, obtemos o seguinte:

Teorema 5.4.10. *Seja φ uma sentença em Sent. São equivalentes:*
(1) φ é válida (com relação à semântica em álgebras de Heyting).
(2) $v^I_{can}(\varphi) = \top$.
(3) φ é teorema de PI.

Definindo o conceito de inferência semântica $\Gamma \models_{\mathcal{H}} \varphi$ numa álgebra de Heyting $\mathcal{H}$ e o conceito de inferência semântica $\Gamma \models_{AH} \varphi$ na classe de álgebras de Heyting de maneira análoga às definições feitas com álgebras de Boole, obtemos o seguinte resultado:

Corolário 5.4.11. *Seja $\Gamma \cup \{\varphi\}$ um subconjunto finito de Sent. São equivalentes:*
(1) $\Gamma \models_{AH} \varphi$.
(2) $\bigwedge\{v^I_{can}(\gamma) \ : \ \gamma \in \Gamma\} \leq v^I_{can}(\varphi)$.
(3) $\Gamma \vdash_{\mathsf{PI}} \varphi$.

Assim, testar conseqüência semântica em *todas* as álgebras de Heyting equivale a testar conseqüência semântica com relação à valoração canônica na álgebra de Lindembaum-Tarski. Mas agora não temos uma álgebra de Heyting canônica (como $\mathcal{B}_\mathbf{2}$) para testar a conseqüência semântica da lógica intuicionista. Esta é uma importante diferença entre a lógica clássica e a intuicionista, relacionada com uma importante diferença entre as álgebras de Boole e as álgebras de Heyting.

Observação 5.4.12. Para obter um sistema de dedução natural para a LPI, basta eliminar a regra

$$(\mathbf{f}2)\ \frac{\begin{array}{c}[\neg\varphi]\\ \mathbf{f}\end{array}}{\varphi}$$

do sistema DN, obtendo um sistema de prova DNI equivalente a PI, sendo portanto adequado para a semântica em álgebras de Heyting.

Finalmente, a partir dos resultados algébricos obtidos neste capítulo, obtemos o seguinte resultado, cuja prova deixamos como exercício para o leitor:

Proposição 5.4.13. *O sistema axiomático PI^+ obtido de PI acrescentando o axioma*

(Axio11) $\neg\varphi \vee \varphi$

é uma axiomatização correta e completa da LPC.

Vemos portanto que a algebrização de uma lógica (no caso, a LPC ou a LPI) fornece mais uma ferramenta para analisar a relação de conseqüência $\vdash$ da lógica: para determinar se $\gamma_1, \ldots, \gamma_n \vdash \varphi$ basta analisar se $a_1 \wedge \cdots \wedge a_n \leq a$, em que a_i é um termo (da linguagem das álgebras associadas à lógica) que corresponde a γ_i, e analogamente para a com relação a φ. Assim, a relação $\vdash$ transforma-se na verificação de $\leq$ numa classe de reticulados apropriados. Por exemplo, testar que $\varphi \vdash_{\mathsf{PI}} \neg\neg\varphi$ equivale a provar que $a \leq - - a$ em toda álgebra de Heyting. Assim, um bom conhecimento dos reticulados associados a uma lógica pode servir como auxiliar para resolver questões sobre derivabilidade nessa lógica. Porém, deve ser observado que, frequentemente, os modelos algébricos (e os métodos semânticos em geral) são mais úteis para provar *não*-derivabilidade do que para provar derivabilidade. Assim, para determinar que $\Gamma \nvdash_{\mathsf{PI}} \varphi$ procura-se provar que $\Gamma \not\models_{AH} \varphi$ através de uma álgebra de Heyting $\mathcal{H}$ e uma valoração v sobre $\mathcal{H}$ apropriada. Analogamente, para determinar que $\Gamma \nvdash_{\mathsf{PC}} \varphi$ procura-se provar que $\Gamma \not\models_{AB} \varphi$ através de uma álgebra de Boole $\mathcal{B}$ e uma valoração v sobre $\mathcal{H}$ apropriada.

5.5 Exercícios

1. Provar que, em $\mathsf{Sent}/_{\approx}$, temos que $[\varphi \wedge \psi] = \bigwedge\{[\varphi], [\psi]\}$.

2. Provar as proposições 5.3.4 e 5.3.5.

3. Provar as proposições 5.3.7 e 5.3.9.

4. Provar a Proposição 5.3.13.

5. Provar que o pseudo-complemento relativo $a \sqsupset b$ satisfaz, para todo $c \in A$:
$$c \leq a \sqsupset b \text{ sse } c \sqcap a \leq b.$$

6. Provar a Proposição 5.3.22.

7. Seja $\langle \mathbb{R}, \leq \rangle$ o poset dos números reais com a sua ordem usual. Considere a relação $x < y$ sse $x \leq y$ e $x \neq y$. Para todo $a, b \in \mathbb{R}$ definimos o *intervalo aberto* (a, b) como sendo o conjunto $(a, b) = \{x \in \mathbb{R} \ : \ a < x < b\}$. Note que $(a, b) = \emptyset$ sse $b \leq a$. Em particular, $(a, a) = \emptyset$. Seja $IA(\mathbb{R})$ o conjunto de todos os intervalos abertos de $\mathbb{R}$. Dizemos que $U \subseteq \mathbb{R}$ é *aberto* se existe $Z \subseteq IA(\mathbb{R})$ tal que $U = \bigcup Z$. O conjunto $\Omega(\mathbb{R})$ dos subconjuntos abertos de $\mathbb{R}$ é a *topologia usual* da reta real. Dado um conjunto $X \subseteq \mathbb{R}$, definimos o *interior* de X com sendo o conjunto $X^\circ = \bigcup\{U \in \Omega(\mathbb{R}) \ : \ U \subseteq X\}$. Com estas definições, provar o seguinte:
(a) $\Omega(\mathbb{R})$ é uma topologia sobre $\mathbb{R}$, isto é: (i) $\mathbb{R}$ é aberto; (ii) se $Y \subseteq \Omega(\mathbb{R})$ então $\bigcup Y \in \Omega(\mathbb{R})$, e em particular $\emptyset$ é aberto (*a união de uma coleção arbitrária de abertos é aberta*); (iii) se $U, V \in \Omega(\mathbb{R})$ então $U \cap V \in \Omega(\mathbb{R})$ (*a interseção de uma coleção finita de abertos é aberta*).
(b) $U \subseteq \mathbb{R}$ é aberto sse para todo $x \in U$ existe $(a, b) \in IA(\mathbb{R})$ tal que $x \in (a, b)$ e $(a, b) \subseteq U$.
(c) Para todo $X \subseteq \mathbb{R}$ temos que X° é o maior aberto contido em X, isto é: (i) $X^\circ \in \Omega(\mathbb{R})$ e $X^\circ \subseteq X$; (ii) se $U \in \Omega(\mathbb{R})$ e $U \subseteq X$ então $U \subseteq X^\circ$.
(d) $\langle \Omega(\mathbb{R}), \subseteq \rangle$ é um reticulado tal que, para todo $U, V \in \Omega(\mathbb{R})$:
$$U \sqcup V := U \cup V \ \text{ e } \ U \sqcap V := U \cap V.$$
(e) $\langle \Omega(\mathbb{R}), \subseteq \rangle$ é um reticulado relativamente pseudo-complementado tal que, para todo $U, V \in \Omega(\mathbb{R})$:
$$U \sqsupset V := \bigcup\{W \in \Omega(\mathbb{R}) \ : \ W \cap U \subseteq V\}.$$
(f) $\langle \Omega(\mathbb{R}), \subseteq \rangle$ é uma álgebra de Heyting tal que, para todo $U \in \Omega(\mathbb{R})$:
$$-U := (\mathbb{R} - U)^\circ.$$

8. Provar a Proposição 5.3.24. Para provar os items (1), (4), (5) e (7) use o exercício anterior.

9. Provar a Proposição 5.3.26.
10. Provar a Proposição 5.4.1
11. Provar a Proposição 5.4.13.

Referências Bibliográficas

[1] E.W. Beth. *The Foundations of Mathematics.* North-Holland, 1959.

[2] S. Burris and H. Sankappanavar. *A Course in Universal Algebra.* Springer-Verlag, New York, 1981. Última edição disponível em http://www.thoralf.uwaterloo.ca/htdocs/ualg.html.

[3] W.A. Carnielli and M. Rathjen. Hydrae and subsystems of arithmetic. Int. Bericht Institut für Math. Logik und Grudlagenforschun, Univ. Münster, 1991.

[4] N. C. A. da Costa. *Sistemas Formais Inconsistentes.* PhD thesis, Universidade Federal do Paraná, Curitiba, Brasil, 1963. Editora UFPR, Curitiba, 1993.

[5] R.L. Epstein. *Propositional Logics: The semantic foundations of logic,* with the assistance and collaboration of W.A. Carnielli, I.M.L. D'Ottaviano, S. Krajewski, and R.D. Maddux. Wadsworth-Thomson Learning, 2^{nd} edition, 2000.

[6] G. Gentzen. Untersuchungen über das logische schliessen. *Mathematische Zeitschrift*, 39:176–210/405–431, 1934.

[7] K.J.J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.

[8] S. Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica*, 1:5–32, 1934.

[9] I. Johánsson. Der minimalkalkül, ein reduzierter intuitionistischer formalismus. *Compositio Mathematica*, 4(1):119–136, 1936.

[10] E. Mendelson. *Introduction to Mathematical Logic.* International Thomson Publishing, 4^{th} edition, 1997.

[11] D. Prawitz. *Natural Deduction.* Almqvist and Wiksell, Stockholm, 1965.

[12] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics.* Polish Scientific Publishers, Warszawa, 2^{nd} edition, 1968.

[13] R.M. Smullyan. *First-Order Logic.* Springer-Verlag, 1968.

[14] R. Wójcicki. *Theory of Logical Calculi.* Synthese Library. Kluwer Academic Publishers, 1988.