Lógic∀ ∃xiste para todos
um mínimo de lógica e argumentação

Walter Carnielli, Marcelo E. Coniglio e Juliana Bueno-Soler
Departamento de Filosofia e
Centro de Lógica, Epistemologia e História da Ciência
Universidade Estadual de Campinas
C.P. 6133, CEP 13081-970
Campinas, SP, Brasil
Centro de Ciências Naturais e Humanas
Universidade Federal do ABC
Santo Andrá, SP, Brasil

(Críticas, comentários e sugestões são bem-vindos)

7 de março de 2013

Sumário

Capítulo 1

Os Paradoxos e seu Significado no Pensamento

1.1 Paradoxos Lógicos e a Questão do 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.

Devemos a Aristóteles a distinção fundamental entre o infinito atual e o infinito potencial, como uma noção de totalidade a ser indefinidamente completável, e o infinito atual, uma totalidade infinita completada, 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).

Segundo Aristóteles, aos matemáticos bastava a noção de infinito potencial, opinião que ainda continuou a ser sustentada por H. Poincaré, no século XX. Essa idéia, contudo, não corresponda à realidade da prática matemática, já que a noção de infinito atual é essencial a muitas teorias matemáticas.

O infinito atual apenas recebeu um tratamento matemático apropriado com a teoria dos conjuntos de Cantor no século XIX, embora suas idéias tenham sido bastante criticadas e incompreendidas. 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 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.

Vamos ver a seguir vamos alguns tipos de propriedades e de problemas que ilustram 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 O infinito paradoxal

1.2.1 O Paradoxo de Galileu, ou a confusão entre as partes e o todo

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 infinitos infinitos

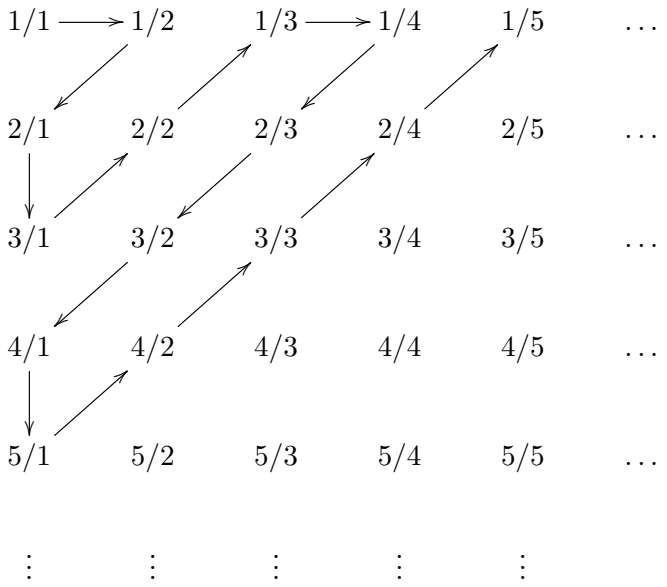
Durante muito tempo (até pelo menos o século XIX) pensou-se que os números racionais não poderiam ser enumeráveis (ou contáveis) como os naturais, uma vez que entre cada dois racionais $\frac{a}{b}$ e $\frac{c}{d}$ existe sempre outro, que é sua média aritmética:

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

e portanto entre dois racionais há infinitos outros.

Georg Cantor mostrou por um artifício simples e genial 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:



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*).

É possível dar uma função muito simples que calcula exatamente a posição de cada fração na lista: dada a fração $\frac{m}{n}$ sua posição será dada por:

$$J(m,n) = \frac{1}{2}[(m+n)(m+n+1)] + m$$

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}	$\ldots$	a_{1n}	$\ldots$
2	0,	a_{21}	a_{22}	a_{23}	a_{24}	$\ldots$	a_{2n}	$\ldots$
3	0,	a_{31}	a_{32}	a_{33}	a_{34}	$\ldots$	a_{3n}	$\ldots$
4	0,	a_{41}	a_{42}	a_{43}	a_{44}	$\ldots$	a_{4n}	$\ldots$
$\vdots$					$\vdots$			
m	0,	a_{m1}	a_{m2}	a_{m3}	a_{m4}	$\ldots$	a_{mn}	$\ldots$
$\vdots$					$\vdots$			

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:

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

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.

O 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: como a intuição nos engana

Uma brincadeira instigante que ilustra muito bem como nos devemos acautelar quando se trata do infinito, e como podemos errar redondamente ao tentar manter nossa intuição acostumada ao mundo dos fatos nesse mundo virtal do infinito, é o chamado "Hotel de Hilbert": existe um certo hotel, com um número infinito e enumerável de quartos

$$Q_1, Q_2, Q_3, \ldots Q_n, \ldots$$

Por sorte do proprietário, os quartos estão todos lotados.

Devemos, é claro, supor que este hotel existiria num mundo possível onde o número de habitantes também fosse infinito. 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 desespera. 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]

[1] Os conceitos formais de *árvore*, *ramo*, *descendente*, *nós* e *nós sucessores* serão definidos posterirmente neste livro.

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

   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 valor e 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

A Lógica Proposicional – Linguagem e Semântica

Neste capítulo são apresentados os conceitos básicos para a construção de sistemas formais, enfatizando a diferença entre linguagem e metalinguagem. O objetivo do capítulo é mostrar ao leitor o processo de construção de um sistema formal, enfatizando.

Os conceitos que são tratados neste capítulo são essencias para a compreensão e a leitura dos demais capítulos. Os exercícios deste capítulo têm o objetivo de auxiliar a compreensão das definições e dos conceitos tratados aqui.

Introdução

Apesar de diversos autores e obras tentarem apresentar a lógica como uma teoria do raciocínio, os paradoxos e antinomias, como vimos no Capítulo 1, mostram quanto é difícil dominar a razão, e quanto as opiniões pré-concebidas nos levam ao engano. Mais ainda, como já sugerimos, e como veremos mais tarde, há de fato limites para o conhecimento e para o alcance do raciocínio.

Inseri neste ponto a definição de lógica matemática, uma noção particular de lógica e que é de maior interesse para este material.

Pelo que vimos também no Capítulo **??**, a Lógica é o único instrumento pelo qual podemos avaliar argumentos de forma global, dado que a Lógica consiste num aparato *a priori* e portanto, universal. É mais interessante por isso, e para nossos propósitos, considerar a lógica como uma teoria da *comunicação* ou *exposição* 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 ...". Nesse mesmo sentido, podemos considerar a *Lógica Matemática* como uma teoria a respeito do tipo de ar-

gumentação feita pelos matemáticos, ou seja, o estudo do tipo de raciocínio realizado pelos matemáticos. Para compreender a lógica matemática é necessário examinar os métodos usados pelos matemáticos.

Vimos também como a forma lógica de certo modo realiza este aparato *a priori* e trabalha a nosso favor, permitindo que passemos de contextos obscuros a outros mais claros quando em dúvida sobre a validade de um argumento: por um lado, podemos construir contra-exemplos eloquentes e definitivos contra argumentos inválidos, graças à forma lógica, e por outro podemos ter já um arsenal de argumentos cristalinamente válidos por meio de demonstrações, de novo graças à forma lógica. Precisamos então nos preocupar com três tarefas básicas:

Neste ponto ser mais preciso quanto ao desenvolvimento do sistema formal, explicar o motivo que o desenvolvimento é feito desse modo.

1. Especificar uma *linguagem simbólica* para que possamos expressar argumentos de forma econômica e universal;

2. Esclarecer os mecanismos que caracterizem argumentos, separando os *argumentos válidos* dos *inválidos*; e

3. Definir as noções de *provas* ou *demonstrações*, isto é, as seqüências de argumentos que possam ser vistos como universalmente válidos.

Falar um pouco sobre os conceitos de proposição, argumento, sentença de modo a fazer uma ponte com a linguagem natural enfatizando as vantagens de se trabalhar com uma linguagem artificial.

Como vimos, devemos nos ater às sentenças ditas *declarativas*, evitando assim sentenças interrogativas, temporais, poéticas, opinativas, bem como súplicas, ordens, suspiros, intenções, etc.

A rigor, podemos também considerar no rol das sentenças declarativas as sentenças *performativas*, usadas por exemplo em linguagens computacionais, que são também sentenças matemáticas, mas tais sentenças performativas podem ser interpretadas (ou seja, reescritas) como sentenças declarativas.

Precisamos obter uma linguagem precisa, que possa ela mesma ser *objeto* de exame rigoroso. Iniciamos nossa análise com os argumentos que enolvem *proposições* estudando assim a *lógica proposicional clássica*, ou *cálculo proposicional*, ou ainda *cálculo sentencial* a qual denotaremos por LPC.

Mais tarde nossa linguagem será expandida 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.*

A linguagem rigorosa e os métodos abstratos que iremos introduzir realizam a ideia da forma lógica que vimos discutindo, no sentido de se caracterizar a Lógica como um instrumento independente para o conhecimento como queria Kant:

> ...uma ciência *a priori* das leis necessárias do pensamento, não, porém, relativamente a objetos particulares, mas a todos os objetos em geral...

2.1 Linguagens proposicionais

2.1.1 Uso, menção e confusão

Uma dstinção que pode dar bastante complicações se não a temos clara é a diferença entre uso e menção, que é essencial em Lógica. Esta é a diferença entre usar uma expressão lingüística para se *referir* a alguma coisa (ou pessoa) e *mencionar*, isto é, falar a respeito dessa coisa (pessoa). Quando mencionamos (ou falamos sobre) um objeto, em geral usamos uma expressão lingüística que nomeia ou de algum modo se refere ao objeto. Como regra, o nome ou a expressão usada não coincide com a coisa a que se refere, e essa coisa por sua vez é mencionada ao se usar seu nome. Quando falamos sobre Sócrates, por exemplo, obviamente usamos a expressão 'Sócrates' e não o próprio Sócrates corporalmente. Desse modo, na sentença:

(1) O Brasil é pequeno.

a palavra 'Brasil' está sendo usada para falar do Brasil, afirmando que ele é pequeno (note que não importa absolutamente aqui se a sentença é verdadeira ou falsa). Por outro lado, na sentença

(2) 'Brasil' tem 6 letras.

não mais estamos falando do Brasil, mas da expressão que nomeia o pais– usamos o nome do país e não o próprio país: em (1) a expressão 'Brasil' está sendo usada, e em (2) está sendo mencionada.

Segundo uma convenção estabelecida pelo lógico e filósofo Willard Van Ormann Quine em *Mathematical Logic (1979)*, a sintaxe para nomear uma expressão (palavra, frase ou sentença) é colocá-la entre aspas simples. Assim, podemos dizer que:

(3) O nome de 'Brasil' é ''Brasil''.

o que é totalmente diferente de dizer:

(4) O nome do Brasil é 'Brasil'.

Mas podemos continuar, e criar nomes para nomes de nomes, e assim por diante:

(5) O nome de ''Brasil'' é '''Brasil'''.

Ou ainda:

(6) O nome do nome do Brasil é ''Brasil''.

A "confusão uso/menção" consiste em usar uma palavra ou expressão quando ela devia estar sendo nomeada, e nomeá-la quando ela está sendo usada.

Na página 24 de *Mathematical Logic* Quine explica:

> Para mencionar Boston, usamos 'Boston' ou um sinônimo, e para mencionar 'Boston' usamos ''Boston'' ou um sinônimo. ''Boston'' contém seis letras e um par de aspas simples; 'Boston' contém seis letras e não tem aspas. E Boston contém cerca de 800.000 pessoas.

No primeiro caso está escrito ''Boston'', e ele está mencionando a expressão 'Boston'. No segundo caso está escrito 'Boston', e ele está mencionando a expressão Boston. No terceiro caso, está escrito Boston , e ele está usando o que a expressão Boston denota, a saber, a cidade de Boston.

A importância desse cuidado em Lógica aparece quando aplicamos o critério (ou definição) de verdade. Ao classificar uma sentença como verdadeira ou falsa, não estamos usando a sentença, mas mencionando-a, como por exemplo:

(7) 'O Brasil é pequeno' é falsa.

Um outro exemplo, onde isso fica ainda mais sutil, é o famoso exemplo:

(8) 'A neve é branca' é verdadeira se e somente se a neve é branca.

significando que 'A neve é branca' nomeia uma expressão verdadeira se e somente seu uso é coerente, ou seja, aquilo que entendemos como neve é alguma coisa de fato branca.

2.1.2 Linguagem, metalinguagem, variáveis e metavariáveis

No Capítulo **??** já mencionamos a distinção entre *linguagem objeto* e *metalinguagem*. A linguagem-objeto é aquela que queremos estudar, para as quais as definições serão propostas, a qual propomos esclarecer, e a respeito da qual pretendemos conhecer. A metalinguagem, como dissemos, é o ambiente onde isso pode ser realizado.

Na verdade, estamos aqui usando os conceitos de linguagem em níveis diferentes, e daí a distinção dada pelo prefixo "meta". A ideia não é muito distinta daquela entre uso e menção, a não ser que aqui se trata de um uso global para a linguagem, e de uma menção global para a metalinguagem. Poderemos, se for necessário, pensar na metametalinguagem, mas isso só seria tentado se houvesse necessidade.

A linguagem permite introduzir de maneira muito intuitiva a noção de *variáveis*: mesmo a linguagem natural permite os pronomes, que são uma espécie de "buracos" a ser preenchidos com nomes adequados. Se digo, por exemplo, "Ele está furioso" não faz sentido trocar "ele", por exemplo, pelo

numeral '5'. Faz sentido, no caso, trocar "ele" pelo *nome* de uma pessoa do sexo masculino. Só assim a sentença seria candidata a poder ser classificada como verdadeira ou falsa; caso contrário, sequer faz sentido.

As variáveis são tão importantes, que praticamente a ciência moderna deve a elas sua vitalidade: não fossem as variáveis terem sido introduzidas na álgebra e depois no cálculo diferencial e integral, a história da ciência seria outra.

Os geômetras gregos, por exemplo, não dominavam a álgebra, e empregavam a geometria para resolver, por processos por vezes complicados, problemas de fácil solução algébrica. É curioso que na Matemática somente no século III Diofante de Alexandria tenha introduzido o uso de símbolos e sinais para denotar quantidades e operações, mas que na Lógica, muito antes Aristóteles já tivesse usado variáveis para expressar propriedades da sua lógica.

Passando de pronomes a variáveis, no caso por exemplo da sentença $x + 2 = 2 + x$, só poderemos avaliá-la se trocarmos x por um numeral, já que se trocarmos por 'Paulo' deixará de ter sentido.

Por outro lado, na sentença lógica $A \to B$ talvez possamos substituir A por 1 e B por 2, *se* 1 e 2 forem *nomes* cujos *valores* forem expressões lógicas.

Temos que ter sempre claro então quais são os chamados *sustituendos*, isto é, quais são as expressões que podem ser colocadas no lugar das variáveis num certo contexto, e quais são os valores que uma variável pode tomar.

Do mesmo modo como usamos variáveis com relação à linguagem, usamos metavariáveis com relação à metaliguagem. Por exemplo, A e B são variáveis que variam sobre sentenças lógicas em $A \to B$, mas poderíamos dizer que α é uma metavariável que varia sobre sentenças lógicas do tipo $A \to B$, que contêm uma implicação.

2.1.3 Enunciados e proposições

Dado que em Lógica, para avaliar argumentos, usamos sentenças da linguagem formal ou da linguagem natural, temos que ter cuidado com armadilhas semânticas como por exemplo com a sentença:

(1) "Sou chinês"

Dita por um chinês, ela será verdadeira; por outro lado, a sentença dita por mim:

(2) "Não sou chinês"

também será verdadeira, e teríamos um problema grave:

(3) "Sou chinês e não sou chinês" seria verdadeira?

Ou ainda, (1) e (2) seriam verdadeiras e falsas ao mesmo tempo, se ditas por um chinês e um não-chinês?

A questão é que (1) e (2) são *enunciados* que expressam *proposições* diferentes segundo quem as profere. Estes casos são dos mais complicados, por causa da presença do chamado *indéxico* (o termo "eu", oculto no verbo). Mas há casos mais simples:

(4) "3 divide 111"

e

(5) "111 é divisível por 3"

são claramente enunciados distintos que expressam a mesma relação entre 111 e 3.

O conteúdo da relação, a qual podemos classificar como verdadeira ou falsa, é o que chamaremos de *proposição*.

O que a Lógica Proposicional possibilita é dar nomes simbólicos para todas as proposições (mas não para os enunciados) e a partir daí trabalhar com as combinações de proposições através dos conectivos. Estes nomes simbólicos para proposições e suas combinações é que constituirá a linguagem- objeto da Lógica Proposicional, que é o que estudaremos a seguir.

2.1.4 Assinaturas e linguagens

Começamos especificando a linguagem-objeto da Lógica, através da técnica das chamadas *definições indutivas*, que consiste em apresentar primeiro as proposições atômicas (isto é, que não são decomponíveis em sentenças mais simples) e, num segundo estágio, especificar como as sentenças mais complexas são construídas a partir das menos complexas utilizando *conectivos*. Os conectivos, como o nome diz, 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 $(\varphi \wedge \psi)$, e o mesmo para a *disjunção* $\vee$, a *implicação* $\rightarrow$ e o *bicondicional* $\leftrightarrow$.

Usualmente os conectivos são unários ou binários, isto é, de uma ou duas entradas,[1] mas é possível definir linguagens formais onde podemos ter conectivos de qualquer número de entradas. A negação é unária, e a implicação, disjunção, conjunção e bicondicional são binários. Mas podemos

[1] Ao invés de "entradas", usa-se também "argumentos", como "$f(x, y) = x + y$ é uma função de dois argumentos". Preferimos, contudo, evitar confundir isso com nossa noção de argumento.

pensar num conectivo ternário do tipo "se α então β, caso contrário γ" que relaciona três entradas.[2] No caso da Lógica Proposicional, porém, bastam os conectivos unários e binários: mesmo este ternário pode ser escrito com combinação dos outros– você é capaz de explicar como?

Chegamos assim à definição seguinte:

Definição 2.1.1. Uma *assinatura proposicional* é um conjunto C de conectivos. △

A assinatura pode ser, por exemplo, $C_1 = \{\wedge, \neg\}$ ou $C_2 = \{\vee, \neg\}$, ou ainda $C_3 = \{\rightarrow, \neg\}$, ou todos juntos: $C_4 = \{\wedge, \rightarrow, \vee, \leftrightarrow, \neg\}$. Como veremos, com a negação e um entre $\{\wedge, \rightarrow, \vee\}$ podemos definir todos os outros conectivos.

A idéia da construção de linguagens formais proposicionais é a seguinte: partindo de um conjunto infinito fixado $\mathcal{V} = \{p_1, p_2, \ldots, p_n, \ldots\}$ de símbolos, chamados de variáveis proposicionais, ou de fórmulas (ou sentenças) atômicas, construir as fórmulas mais complexas utilizando os conectivos de uma assinatura C, junto com alguns símbolos auxiliares (parênteses e vírgulas). O uso destes símbolos auxiliares apenas serve para facilitar a individualização das componentes de uma fórmula, sendo, de fato, prescindíveis. Essa linguagem dependerá da assinatura C

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 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 2.1.2. A *linguagem gerada por uma assinatura* C é o conjunto $L(C)$ definido satisfazendo as seguintes propriedades:

[2]Para quem é iniciado em linguagens de programação, este é o conectivo "if-then-else" de certas linguagens.

1. As variáveis proposicionais de $\mathcal{V}$ fazem parte da linguagem gerada;

2. se $c_1 \in C$ é um conectivo unário e φ faz parte da linguagem, então $c_1(\varphi)$ faz parte da linguagem;

3. se $c_2 \in C$ é um conectivo binário e φ_1, φ_2 fazem parte da linguagem, então $c_2(\varphi_1, \varphi_2)$ faz parte da linguagem;

4. nada mais faz parte da linguagem. $\triangle$

Note que usamos na terceira cláusula a expressão $c(\varphi_1, \varphi_2)$ para denotar a aplicação de um conectivo. Tudo se passa como se escrevêssemos, por exemplo, $\vee(\varphi_1, \varphi_2)$ ou $\wedge(\varphi_1, \varphi_2)$ ao invés de $(\varphi_1 \vee \varphi_2)$ ou $(\varphi_1 \wedge \varphi_2)$. Esta última notação, que usamos mais frequentemente, é chamada *infixa*, quando escrevemos o conectivo *entre* os argumentos. A outra é a chamada *prefixa*, quando escrevemos o conectivo *na frente* dos argumentos.

A última cláusula "Nada mais faz parte da linguagem" é bastante pedante: na verdade, ela está aí por culpa da ideia de condicional (na metalinguagem). A rigor, esta definição e composta por cláusulas na metalinguagem do tipo "se ... então"; se algo cumpre o antecedente, então deve cumprir o consequente. Se não cumpre, nada se pode afirmar. Portanto, em princípio algo que não cumpre o antecedente, como a Constelação do Cruzeiro do Sul, por exemplo, poderia, em última instância, fazer parte da linguagem! Para evitar isso, para sermos rigorosos devemos então adicionar a cláusula pedante.
A seguir exemplificaremos as definições introduzidas até agora, definindo três assinaturas especiais que utilizaremos ao longo deste livro para analisar a lógica proposicional clássica.

Freqüentemente economizaremos parênteses, omitindo os mais externos. Assim, escrevemos $\varphi \vee \psi$ ao invés de $(\varphi \vee \psi)$ quando a leitura não for comprometida. Também escreveremos $\neg\varphi$ no lugar de $\neg(\varphi)$ quando não houver risco de confusão.

Exemplo 2.1.3. A assinatura C_0 consiste dos seguintes conectivos:

$$C_0 = \{\neg, \rightarrow\} \quad \text{(negação e implicação)}.$$

As seguintes expressões são fórmulas escritas na assinatura C_0:

$$\neg(p_2 \rightarrow \neg p_1) \quad \text{e} \quad (\neg p_1 \rightarrow p_1) \rightarrow p_1.$$

$\triangle$

Exemplo 2.1.4. A assinatura C_1 consiste dos seguintes conectivos:

$$C_1 = \{\neg, \vee\} \quad \text{(negação e disjunção).}$$

Por exemplo, as seguintes são fórmulas escritas na assinatura C_1:

$$\neg(p_3 \vee p_7) \quad \text{e} \quad p_1 \vee (\neg p_1 \vee p_2).$$

△

Exemplo 2.1.5. A assinatura C_2 consiste dos seguintes conectivos:

$$C_2 = \{\neg, \vee, \wedge, \rightarrow\} \quad \text{(negação, disjunção, conjunção e implicação).}$$

As expressões $p_1 \rightarrow p_1$, $p_4 \rightarrow (p_4 \vee p_5)$ e $\neg p_6 \rightarrow (p_3 \vee (p_3 \wedge p5))$ são fórmulas sobre C_2. Observe que, sem omitir parênteses, as fórmulas acima deveriam ser escritas como

$$(p_1 \rightarrow p_1), \quad (p_4 \rightarrow (p_4 \vee p_5)) \quad \text{e} \quad (\neg(p_6) \rightarrow (p_3 \vee (p_3 \wedge p5)))$$

respectivamente. △

Cada uma das assinaturas acima é suficiente para descrever a lógica proposicional clássica (abreviada por LPC). O conjunto de todas as fórmulas sobre C_1, a assinatura que usaremos daqui por diante, será chamado de Prop.

Observação 2.1.6. Podemos agora formalizar a discussão que fizemos anteriormente sobre liguagem e metalinguagem. As letras $\varphi_1, \ldots, \varphi_n$ utilizadas para se referir a 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 metavariáveis de fórmulas. Reservamos $p_1, p_2, p_3, \ldots$ para denotar as variáveis proposicionais propriamente ditas.

Um conceito útil a ser definido sobre o conjunto Prop é uma função *Var*, que associa a cada fórmula φ o conjunto das variáveis proposicionais que ocorrem na fórmula.

Como estamos tratando da lógica simbólica, tudo pode ser expresso através de conjuntos. Como veremos no Capítulo YY, baseando-se apenas na noção de conjuntos podemos explicitar a noção de função, que desempenha o papel de uma *regra unívoca*, isto, é, uma regra de escolha que nunca "emparelha" um elemento da entrada com mais de um elemento da saída.

Por exemplo, a regra que associa a cada pessoa sua mãe é unívoca, e portanto é uma função; já a regra que associa a cada pessoa sua avó não é unívoca (há duas!) e portanto não é uma função. Usaremos intuitivamente aqui os conceitos de conjunto, conjunto das partes de um conjunto, e de função; usaremos o símbolo $\wp(X)$ para denotar o conjunto das partes do conjunto X, isto é,

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

que é lido como "o conjunto de todos os Y tais que são subconjuntos de X".

Definição 2.1.7. Definimos a função $Var : \mathsf{Prop} \to \wp(\mathcal{V})$, que relaciona fórmulas com conjuntos de variáveis, como segue:

1. $Var(p_i) = \{p\}$ se $p_i \in \mathcal{V}$;
2. $Var(c_n(\varphi_1, \varphi_2, \cdots \varphi_n)) = \bigcup_{i=1}^{n} Var(\varphi_i)$, para c_n um conectivo n-ário.

△

Outra noção útil é a de *subfórmula* de uma fórmula também pode ser definida indutivamente:

Definição 2.1.8. Definimos a função *conjunto de subfórmulas* como sendo a função $SF : \mathsf{Prop} \to \wp(\mathsf{Prop})$, que relaciona fórmulas com conjuntos de fórmulas, como segue:

1. $SF(p_i) = \{p_1\}$, se $p_i \in \mathcal{V}$;
2. $SF(c_n(\varphi_1, \varphi_2, \cdots \varphi_n)) = \{\bigcup_{i=1}^{n} SF(\varphi_i)\} \cup \{c_n(\varphi_1, \varphi_2, \cdots \varphi_n)\}$. △

Os elementos de $SF(\varphi)$ são fórmulas, chamadas *subfórmulas de* φ.

Exemplo 2.1.9. 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 $(p_{10} \vee \neg p_3) \vee \neg p_2$. △

2.2 A semântica da Lógica Proposicional

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, e 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").

A diferença aqui é no fundo a distinção entre uso e menção: se apenas utilizamos os símbolos $\vee$ e $\neg$ sem esclarecer seu sentido, os estamos mencionando. A partir do momento que necessitamos da sua interpretação, os estamos usando.

A partir de agora, denotaremos por $\mathbf{2}$ 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 tabelas abaxo. às vezes é conveniente usar a notação infixa para os operadores binários, escrevendo o operador *entre* os argumentos, e outras vezes a notação prefixa:

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 verdade 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* se satisfaz o seguinte:

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

A ideia a respeito do que segue não é transformar a Lógica numa imensa complicação, mas apenas deixar claro os processos pelos quais a Lógica simbólica trabalha: pretendemos separar as propriedades metalógicas, ou seja, aquelas a respeito da Lógica e que podem ser demonstradas com recursos matemáticos rigorosos, e que chamaremos de *metateoremas*. Indicamos brevemente com quais técnicas os metateoremas podem ser demonstrados, mas isso não será exigido de você (leitor), nem pressupomos que você saiba como fazer. O que você terá que manejar bem serão os teoremas, não os metateoremas. Mas nós acreditamos que mostrar a distinção ajuda bastante.

Metateorema 2.2.2. *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 $\varphi \in \mathsf{Prop}$, ou seja, $v = v'$.*

Demonstração: Por indução na complexidade das fórmulas, seguindo as cláusulas da definição indutiva de valoração.

Suponha, por redução ao absurdo, que existe $\varphi \in \mathsf{Prop}$ tal que $v(\varphi) \neq v'(\varphi)$. Suponha ainda que φ seja a fórmula de menor complexidade tal que isso acontece. De acordo com a Definição 2.1.2, a fórmula φ pode assumir uma das seguintes formas:

1. $\varphi \in \mathcal{V}$.
   Como por hipótese v e v' coincidem em $\mathcal{V}$, então $v(\varphi) = v'(\varphi)$. Absurdo, uma vez que supusemos que $v(\varphi) \neq v'(\varphi)$.

2. $\varphi = \neg\varphi_1$.
   Como φ_1 é uma subfórmula de φ com menor complexidade, temos por hipótese de indução que $v(\varphi_1) = v'(\varphi_1)$. Dado que v e v' são valorações (veja Definição 2.2.1) então $v(\neg\varphi) = -(v(\varphi_1)) = -(v'(\varphi_1)) = v'(\neg\varphi)$. Portanto, $v(\varphi) = v'(\varphi)$. Absurdo.

3. $\varphi = (\varphi_1 \vee \varphi_2)$.
   Como φ_1 e φ_2 são subfórmulas de φ com menor complexidade, temos por hipótese de indução que $v(\varphi_1) = v'(\varphi_1)$ e $v(\varphi_2) = v'(\varphi_2)$. Dado que v e v' são valorações, então $v(\varphi_1 \vee \varphi_2) = \sqcup(v(\varphi_1), v(\varphi_2)) = \sqcup(v'(\varphi_1), v'(\varphi_2)) = v'(\varphi_1 \vee \varphi_2)$. Absurdo.

Como em cada caso temos uma contradição, então é falso supor que existe uma fórmula φ tal que as valorações não coincidam. Portanto $v = v'$. □

Graças ao Metateorema 2.2.2 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 de fato em φ.

Metateorema 2.2.3. *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 das fórmulas. Suponha que o resultado seja válido para toda fórmula em Prop com complexidade menor que φ.

1. $\varphi \in \mathcal{V}$.
   Seja $\varphi = p_i$ para alguma $p_i \in \mathcal{V}$. Dado que $pi \in Var(\varphi)$, por hipótese temos que $v(p_i) = v'(p_i)$.

2. $\varphi = \neg\varphi_1$.
   Dado que φ_1 tem complexidade menor que φ, por hipótese de indução temos $v(\varphi_1) = v'(\varphi_1)$. Daí, pela definição de valoração segue que $v(\neg\varphi_1) = -(v(\varphi_1)) = -(v'(\varphi_1)) = v'(\neg\varphi_1)$.

3. $\varphi = (\varphi_1 \vee \varphi_2)$. Exercício.

Portanto $v(\varphi) = v'(\varphi)$. □

Observação 2.2.4. 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 2.1.5).

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* está 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 o antecedente P e o consequente Q. A interpretação da implicação que é feita em lógica clássica 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 o 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 consequência 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 menção: 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 ideia 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 as 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 fenômeno 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:

$$h(p,q) = \sqcup(-(p), q) \quad \text{e} \quad k(p,q) = -(\sqcup(-(p), -(q)))$$

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_1

(Exemplo 2.1.4) para representar a lógica proposicional clássica (LPC). Voltaremos depois (na Subseção 2.2.4) a falar sobre a questão da definibilidade de funções de verdade em termos de outras funções.

2.2.2 Tautologias, contradições e contingências

Considerando que as proposições são avaliadas através de valorações , o primeiro passo é identificar duas proposições que tomam o mesmo valor em toda valoração .

Definição 2.2.5. Dizemos que duas fórmulas φ e ψ são *semanticamente equivalentes*, e 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 5):

- $\varphi \equiv \varphi$ 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 2.2.6. Temos que $(\varphi \vee \psi) \equiv (\psi \vee \varphi)$ e $\neg\neg\varphi \equiv \varphi$ para toda φ, ψ. △

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

Definição 2.2.7. Seja $\varphi \in$ 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 atribuído 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 do Metateorema 2.2.3, 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 2.2.8. As sentenças $p_1 \vee \neg p_1$, $(p_1 \wedge p_2) \to p_2$ e $p_1 \to (\neg p_1 \to p_2)$ são tautologias:

p_1	$\neg p_1$	$p_1 \vee \neg p_1$
1	0	1
0	1	1

p_1	p_2	$p_1 \wedge p_2$	$(p_1 \wedge p_2) \to p_2$
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	1

p_1	p_2	$\neg p_1$	$\neg p_1 \to p_2$	$p_1 \to (\neg p_1 \to p_2)$
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 $(p_1 \vee p_2) \to p_2$ e $((p_1 \to p_2) \wedge p_2) \to p_1$ não são tautologias:

p_1	p_2	$p_1 \vee p_2$	$(p_1 \vee p_2) \to p_2$
1	1	1	1
1	0	1	0
0	1	1	1
0	0	0	1

p_1	p_2	$p_1 \to p_2$	$(p_1 \to p_2) \wedge p_2$	$((p_1 \to p_2) \wedge p_2) \to p_1$
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 $v(p_1) = 1$ e $v(p_2) = 0$ produz $v((p_1 \vee p_2) \to p_2) = 0$, portanto existem valorações que tornam a fórmula $\varphi = (p_1 \vee p_2) \to p_2$ falsa, logo φ não é tautologia. Por outro lado, toda valoração v tal que $v(p_1) = 0$ e $v(p_2) = 1$ produz $v(((p_1 \to p_2) \wedge p_2) \to p_1) =$

0, portanto esta sentença não é uma tautologia. Esta fórmula representa uma conhecida falácia, a *falácia de afirmação do consequente.* △

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 $\gamma = (\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$ é tautologia. Supomos, por redução ao absurdo, que $v(\gamma) = 0$ para alguma valoração v; então:

1. (a) $v(\varphi \to \psi) = 1$, (b) $v(\neg\psi \to \neg\varphi) = 0$
2. $v(\neg\psi) = 1$, $v(\neg\varphi) = 0$ (de 1(b))
3. $v(\psi) = 0$, $v(\varphi) = 1$ (de 2)
4. $v(\varphi \to \psi) = 0$ (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 2.2.9. Seja $\varphi \in \mathsf{Prop}$. Dizemos que φ é uma *contradição* se $v(\varphi) = 0$ para toda valoração v. $\triangle$

As contradições 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:

Metateorema 2.2.10.
(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 $\psi \equiv \neg\varphi$ é uma contradição, pelo item (i). Mas $\neg\psi \equiv \neg\neg\varphi$ e $\neg\neg\varphi \equiv \varphi$, pelo Exemplo 2.2.6. Como a relação $\equiv$ é transitiva, segue que $\varphi \equiv \neg\psi$, sendo que ψ é uma contradição. $\square$

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 Prop_2 definida a partir da assinatura C_2 (Exemplo 2.1.5). 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.

Acho que seria interessante falar que escrever as fórmulas em FNC ou FND serve para trabalhar com a Regra da Resolução, uma maneira adequada para se tratar computacionalmente.

Antes de passar às definições, introduzimos a seguinte notação:

Notação 2.2.11. 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 2.2.12. (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.

Metateorema 2.2.13. *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 as variáveis proposicionais que ocorrem em φ. 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, deve 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 $v(\varphi)$, pelo Metateorema 2.2.3. 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. □

Esta demonstração é bem complicada, porque na verdade ela é um *algoritmo* que já constrói a FND!

Exemplo 2.2.14. Mostraremos, como exemplo, 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 2.2.15. (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:

Metateorema 2.2.16. *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 do Metateorema 2.2.16 (a partir do Metateorema 2.2.13) é deixada para o leitor como um desafio.

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 do Metateorema 2.2.13. 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 2.2.17. 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.

Metateorema 2.2.18. $\{\neg, \vee\}$ *é um conjunto adequado de conectivos.*

Demonstração: Já está dada acima: a conjunção pode ser representada em termos de $-$ e $\sqcup$ através da equação $p \sqcap q = -(-p \sqcup -q)$. □

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 do Metateorema 2.2.18, podemos provar o seguinte:

Metateorema 2.2.19. $\{\neg, \wedge\}$ *é um conjunto adequado de conectivos.*

Por outro lado, da equação $p \sqcup q = -p \sqsupset q$ e do Metateorema 2.2.18 obtemos imediatamente o seguinte:

Metateorema 2.2.20. $\{\neg, \rightarrow\}$ *é um conjunto adequado de conectivos.*

Consideremos agora uma constante $\perp$ tal que $v(\perp) = 0$ para toda valoração v. Então $-p = p \sqsupset \perp$ (confira!), logo obtemos o seguinte:

Metateorema 2.2.21. $\{\perp, \rightarrow\}$ *é um conjunto adequado de conectivos.*

Por outro lado, provaremos o seguinte:

Metateorema 2.2.22. $\{\neg, \leftrightarrow\}$ *e* $\{\vee, \wedge\}$ *não são conjuntos adequados de conectivos.*

Demonstração: Seja $\varphi(p, q)$ uma função de verdade de duas entradas p e q construída da utilizando apenas as funções $-$ e $\Leftrightarrow$. Podemos provar que φ é tautologia, ou contradição, ou as quatro entradas da tabela-verdade de φ consistem exatamente de dois valores 0 e dois valores 1. Dessa maneira, fica claro que não é possível definir a tabela da disjunção (pois as quatro entradas da sua tabela consistem de três valores 1 e um valor 0) nem a da conjunção (pois as quatro entradas 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 uma entrada 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. □

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:

Metateorema 2.2.23. *O conjunto* $\{$ *NAND* $\}$ *é adequado.*

Similarmente, definimos um conectivo NOR dado por:

- (p 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:

Metateorema 2.2.24. *O conjunto* $\{$ *NOR* $\}$ *é adequado.*

É possível provar o seguinte (veja [6]):

Metateorema 2.2.25. *NAND e NOR são os únicos conectivos adequados.*

2.2.5 Argumentos e consequência semântica

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

Definição 2.2.26. Seja Γ um conjunto de proposições em Prop, e seja $\varphi \in \mathsf{Prop}$. Diremos que φ é *consequência 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 $\Gamma \not\models \varphi$. $\triangle$

Um caso particular interessante da definição anterior ocorre quando $\Gamma = \emptyset$. Nesse casso, $\emptyset \models \varphi$ significa que φ é uma tautologia.
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.

A noção de consequência semântica corresponde precisamente à versão de argumento no caso proposicional:

$$\Gamma \models \varphi$$

se, e somente se, é impossível que exista uma valoracao v que *satisfaça* Γ no sentido em que $v(\psi) = 1$ para todo $\psi \in \Gamma$, e que *rejeite* φ no sentido em que $v(\varphi) = 0$.

Esta noção de inferência (ou de consequência) lógica é precisamente, para a Logica de Proposições, aquela ideia de "... segue de ..." que mencionamos no Capítulo **??**.

Finalmente, todos aqueles argumentos que podem ser expressos em termos somente dos conectivos têm agora uma ferramente completa para que possam ser avaliados.

E quando dizemos "completa", não estamos brincando nem falando meio vagamente–como vamos ver depois, existe um Metateorema de Completude para a Lógica Proposicional que demonstra que a consequência semântica é definitiva: ela basta para decidir todo e qualquer tipo de argumento proposicional!

Exemplos 2.2.27.

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 consequência semântica são as seguintes:

Metateorema 2.2.28. *A relação de consequência $\models$ 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. $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\} \models \varphi$ *see* $\Gamma \models \varphi_1 \rightarrow (\varphi_2 \rightarrow (\varphi_n \rightarrow \varphi) \ldots)$*;*
11. $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\} \models \varphi$ *see* $\Gamma \models (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \varphi$*.*

Demonstração: Provaremos somente alguns itens como ilustração, o restante é deixado para o leitor como exercício.

(3) Seja v uma valoração tal que $v(\Gamma) = 1$, isto é, $v(\gamma_i) = 1$ para todo $\gamma_i \in \Gamma$. Como, por hipótese, $\varphi \in \Gamma$ temos que $v(\varphi) = 1$. Portanto, pela Definição 2.2.26, segue que $\Gamma \models \varphi$.

(6) Suponha, por redução ao absurdo, que $\Gamma, \Delta \nvDash \psi$. Isso significa que existe uma valoração v tal que $v(\Gamma \cup \Delta) = 1$ e $v(\psi) = 0$. Como, por hipótese, $\Gamma \models \varphi$, pela Definição 2.2.26, isso significa que para toda valoração v', se $v'(\Gamma) = 1$ então $v'(\varphi) = 1$. Em particular, dado que $v(\Gamma \cup \Delta) = 1$ e consequentemente $v(\Gamma) = 1$ e $v(\Delta) = 1$, segue que $v(\varphi) = 1$, e daí $v(\Delta \cup \{\varphi\}) = 1$. Como, por hipótese, $\Delta, \varphi \models \psi$, então $v(\psi) = 1$. Absurdo.

(8) ($\Longrightarrow$) Segue de (2).
($\Longleftarrow$) 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, $\models \varphi$.

(9) ($\Longrightarrow$) Seja v uma valoração tal que $v(\Gamma) = 1$. Temos que $v(\varphi) = 1$ ou $v(\varphi) = 0$.

- Se $v(\varphi) = 1$, já que $v(\Gamma) = 1$, então $v(\Gamma \cup \{\varphi\}) = 1$, daí $v(\psi) = 1$ (pois, por hipótese, $\Gamma, \varphi \models \psi$). Portanto, $v(\varphi \to \psi) = 1$.
- Se $v(\varphi) = 0$, claramente $v(\varphi \to \psi) = 1$.

Portanto, para toda valoração v, se $v(\Gamma) = 1$ então $v(\varphi \to \psi) = 1$, isto é, $\Gamma \models \varphi \to \psi$.
($\Longleftarrow$) Suponha que $\Gamma, \varphi \nvDash \psi$, isto é, que existe uma valoração v tal que $v(\Gamma \cup \{\varphi\}) = 1$ e $v(\psi) = 0$. Claramente, para essa valoração, temos que $v(\varphi \to \psi) = 0$. Portanto $\Gamma \nvDash \varphi \to \psi$.

□

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

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 2.2.29. Dada uma variável proposicional p e uma proposição ψ, definimos por recursão uma função *substituição de ψ em lugar de p* como a função $Subs_p^{\psi} : \mathsf{Prop} \to \mathsf{Prop}$ 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. $Subs_p^{\psi}(\neg\varphi_1) = \neg Subs_p^{\psi}(\varphi_1)$. $\triangle$

Escreveremos $\varphi[\psi/p]$ no lugar de $Subs_p^{\psi}(\varphi)$.

Vamos tratar agora de uma importante propriedade da relação de consequência semântica $\models$, a saber: o resultado de substituir componentes atômicos por partes semanticamente equivalentes produz fórmulas semanticamente equivalentes. Por exemplo, sejam

$$\psi_1 = (p_1 \wedge p_2) \quad \text{e} \quad \psi_2 = \neg(p_1 \rightarrow \neg p_2).$$

Como $\psi_1 \equiv \psi_2$, então $\varphi[\psi_1/p] \equiv \varphi[\psi_2/p]$ para qualquer fórmula φ.

Metateorema 2.2.30. (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 $p \in \mathcal{V}$.

$\square$

Demonstração: Por indução na complexidade das fórmulas, seguindo as cláusulas da Definição 2.2.29.

1. Se $\varphi \in \mathcal{V}$

   - Se $\varphi \neq p$, então $\varphi[\psi_1/p] = \varphi = \varphi[\psi_2/p]$.
   - Se $\varphi = p$, então $\varphi[\psi_1/p] = \psi_1 \equiv \psi_2 = \varphi[\psi_2/p]$.

   Logo, em ambos os casos temos que $\varphi[\psi_1/p] \equiv \varphi[\psi_2/p]$.

2. $\varphi = \neg\varphi_1$
   Como φ_1 tem complexidade menor que φ então, pela hipótese de indução, $\varphi_1[\psi_1/p] \equiv \varphi_1[\psi_2/p]$. Isso significa que para toda valoração v (veja Definição 2.2.5) $v(\varphi_1[\psi_1/p]) = v(\varphi_1[\psi_2/p])$. Daí, $v(\neg\varphi_1[\psi_1/p]) = v(\neg\varphi_1[\psi_2/p])$, para toda valoração v. Logo, $\varphi[\psi_1/p] \equiv \varphi[\psi_2/p]$.

3. $\varphi = (\varphi_1 \vee \varphi_2)$
   Exercício.

$\square$

As substituições representam a forma lógica na sua quintessência: substituições serão utilizadas, entre outras coisas, para definir regras de inferência esquemáticas.

Dadas $q_1, \ldots, q_n$ distintos em $\mathcal{V}$, podemos generalizar as funções $Subs_p^{\psi}$ 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 $\varphi[\psi_1/q_1, \cdots, \psi_n/q_n]$ é a substituição *simultânea* enquanto

$$\varphi[\psi_1/q_1][\psi_2/q_2] \cdots [\psi_n/q_n]$$

é a substituição *iterada*, e que em ela são coisas diferentes. Por exemplo, $(p_1 \vee p_2)[p_2/p_1, \psi/p_2] = (p_2 \vee \psi)$ enquanto que

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

É possível (mas muito aborrecido, e nem mesmo os lógicos o fazem) demonstrar o seguinte resultado:

Metateorema 2.2.31. (Teorema da Substituição, versão 2) *Sejam $\psi_1, \ldots, \psi_n, \varphi_1, \ldots, \varphi_n \in$ 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 $\varphi \in$ Prop e para todas $q_1, \ldots, q_n$ distintas em $\mathcal{V}$.*

Com todo este arsenal, já temos instrumentos para saber *tudo* o que se precisa para avaliar proposições como verdadeiras ou falsas.

Na verdade, a maneira de organizar tudo em metateoremas bastante gerais facilita as coisas, porque qualquer embasamento que você buscar para discutir alguma coisa a respeito da avaliação de proposições estará em algum dos metateoremas.

Em resumo, provavelmente tudo o que você vai precisar saber na vida respeito da **semântica** da Lógica Proposicional está aqui...

2.2.6 Exercícios

1. Diga se as sentenças abaixo são verdadeiras ou falsas:
   (a) '5+7' é igual a '12'.
   (b) 'Gödel' demonstrou os Teoremas de Incompletude da lógica.
   (c) "Cantor" não é o nome de Hilbert, mas é o nome de 'Cantor'.
   (d) "A neve é branca' é verdadeira' é uma sentença da português.

(e) Lógica começa com a letra L.

(f) Aristóteles é autor da 'Ética'.

2. Nas sentenças abaixo, caso necessário, coloque aspas a fim de torná-las verdadeiras. Justifique o uso das aspas.

   (a) A sentença George Boole publicou *An Investigation of the Laws of Thought* é falsa.

   (b) $sen^2 + cos^2$ é igual a 1, mas $3 - 2$ é diferente de 1.

   (c) Bertrand Russell é o nome de um grande lógico.

   (d) A palavra function tem o mesmo significado que a palavra portuguesa função.

   (e) Lógica tem 8 letras.

3. Seja $L(C)$ a linguagem gerada pela assinatura C. Para cada assinatura dada, decida se as expressões fazem parte de $L(C)$, justifique em caso negativo.

   (a) $p_1 \to (p_1 \vee p_2)$, para $C = \{\vee, \neg\}$.

   (b) $(p_1 \wedge p_2) \to p_2$, para $C = \{\wedge, \to, \neg\}$.

   (c) $\neg(\neg(p_1 \vee p_2))$, para $C = \{\vee, \wedge, \to, \neg\}$.

   (d) $\to (\neg(p_1 \to p_2))$, para $C = \{\to, \neg\}$.

   (e) $\neg\neg\neg p_3$, para $C = \{\vee, \neg\}$.

   (f) $(p_1 \to p_2) \wedge (p_2 \to p_1)$, para $C = \{\wedge, \to, \neg\}$.

   (g) $(p_1 \leftrightarrow p_2)$, para $C = \{\to, \wedge, \vee, \neg\}$.

   (h) $p_1 \to (\neg p_1 \to p_2)$, para $C = \{\vee, \to, \neg\}$.

4. Para cada fórmula φ, determine o conjunto $Var(\varphi)$ das variáveis que ocorrem em φ:

   (a) $\varphi = (p_1 \vee \neg p_2) \vee (\neg p_1 \wedge p_3)$

   (b) $\varphi = (p_3 \to (p_5 \to p_3))$

   (c) $\varphi = (p_{200} \wedge \neg p_{151}) \wedge p_{21}$

   (d) $\varphi = \neg(p_1 \vee p_2) \vee (p_3 \vee \neg(\neg p_4 \vee p_5))$

   (e) $\varphi = p_1 \to (p_2 \to p_3) \to ((p_1 \to p_2) \to (p_1 \to p_3))$

5. Para cada fórmula φ dada no exercício anterior, determine o conjunto das subfórmulas de φ.

6. Faça a tabela-verdade das seguintes sentenças:

   (a) $[p_1 \to (p_1 \wedge p_2)] \vee p_3$

   (b) $[p_1 \vee (p_2 \to p_3)] \to (p_1 \vee p_3)$

   (c) $p_1 \vee (p_1 \to p_2)$

   (d) $(p_1 \to p_2) \vee (p_2 \to p_1)$

   (e) $(\neg p_1 \wedge p_1) \wedge [(p_2 \vee p_3) \vee \neg(p_2 \vee p_3)]$

7. Seja $v : \mathsf{Prop}_2 \to \{0, 1\}$ uma valoração que satisfaz além das cláusulas da Definição 2.2.1 as seguintes cláusulas indutivas:

   (a) $v(\varphi \wedge \psi) = \sqcap(v(\varphi), v(\psi))$;

   (b) $v(\varphi \to \psi) = \sqsupset (v(\varphi), v(\psi))$

   Demonstre que se v e v' são duas valorações definidas em Prop_2 tais que coincidem em $\mathcal{V}$, então $v(\varphi) = v'(\varphi)$ para toda $\varphi \in \mathsf{Prop}_2$.

8. Seja $\varphi \in \mathsf{Prop}_2$ uma fórmula. Demonstre que se v e v' são duas valorações (definidas como no exercício acima) tais que $v(p) = v'(p)$ para toda $p \in Var(\varphi)$, então $v(\varphi) = v'(\varphi)$.

9. Para cada uma das tabelas abaixo, encontre uma sentença que a determine:

(a)

p_1	p_2	φ
1	1	1
1	0	1
0	1	0
0	0	0

(b)

p_1	p_2	p_3	ψ
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	1

(c)

p_1	p_2	p_3	γ
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	0

10. A partir da existência da FND para toda sentença, prove o Metateorema 2.2.16: toda sentença pode ser reescrita em FNC.

11. Encontre uma sentença em FNC que seja semanticamente equivalente à sentença $(\alpha \to \beta) \vee (\neg\beta \wedge \gamma)$.

12. Verifique se em cada caso a sentença φ é uma consequência semântica de Γ (isto é, $\Gamma \models \varphi$). Em caso negativo, justifique.

    (a) Considere $\varphi = p$ e $\Gamma = \{p,\ (p \to q)\}$

    (b) Considere $\varphi = (p \to r)$ e $\Gamma = \{p \to q,\ q \to r\}$

    (c) Considere $\varphi = r$ e $\Gamma = \{\neg p,\ q \to p, \neg q \to r\}$

    (d) Considere $\varphi = (p \vee q)$ e $\Gamma = \{\neg p,\ \neg q \to r,\ (\neg q \wedge \neg q)\}$

    (e) Considere $\varphi = p$ e $\Gamma = \{p \wedge q\}$

    (f) Considere $\varphi = (p \wedge q)$ e $\Gamma = \{p, q\}$

13. Seja $\{p, q, r, p_1, p_2, p_3\} \subset \mathcal{V}$ e considere as seguintes sentenças:

    - $\varphi = \neg(p \vee q) \vee (\neg p \vee (q \vee r))$
    - $\psi_1 = (p_1 \to p_2) \vee (p \wedge r)$
    - $\psi_2 = \neg(p_1 \wedge \neg p_2) \vee (p \wedge r)$

    Calcule as seguintes substituições:

    (a) $\varphi[\psi_1/p]$
    (b) $\varphi[\psi_1/q]$
    (c) $\varphi[\psi_1/r]$
    (d) $\varphi[\psi_2/p]$
    (e) $\varphi[\psi_2/q]$
    (f) $\varphi[\psi_2/r]$
    (g) $\varphi[\psi_1/p, \psi_2/q]$
    (h) $\varphi[\psi_1/p, \psi_2/r]$
    (i) $\varphi[\psi_2/p, \psi_1/q, \psi_1/r]$
    (j) $\varphi[\psi_1/p, \psi_2/q, \psi_2/r]$

    Dentre as sentenças encontradas acima, determine os pares de sentenças que são semanticamente equivalentes. Justifique sua resposta.

14. Prove que $\varphi \equiv \psi$ sse $\varphi \leftrightarrow \psi$ é tautologia sse $\varphi \to \psi$ e $\psi \to \varphi$ são tautologias sse $\varphi \models \psi$ e $\psi \models \varphi$.

15. Prove que $\{\neg\alpha \to (\psi \vee \beta),\ \neg\gamma,\ \delta \wedge \psi,\ \beta \to \neg\delta\} \models \alpha$.

Capítulo 3

Axiomática e Completude

3.1 Dedução e demonstração

O Dicionário *Houaiss da Língua Portuguesa* ensina que 'demonstração' é sinônimo de 'prova', que seria aquilo que convence que uma afirmação ou um fato são verdadeiros. Usaremos aqui ambos os termos como sinônimos, e ainda como sinônimo de 'dedução', mas com algum cuidado.

Provas ou demonstrações são argumentos especializados, que atuam como formas de comunicação entre interessados. Uma prova em Matemática ou em Lógica é diferente de uma prova num tribunal de justiça: no tribunal, a 'prova do crime' não é um argumento, mas algo real. A noção de prova varia, não somente entre áreas distintas, mas numa mesma área, entre épocas distintas.

Provas em Matemática são as mais sofisticadas, já que tudo em Matemática é demonstração. O método de dedução 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 demonstração, sendo o precursor do que mais tarde viria a ser chamada de *teoria da demonstração* ou *teoria da prova*.

As provas em Lógica Proposicional são muito mais simples, quase um brinquedo comparadas com as da Matemática. O método axiomático hilbertiano em Lógica Proposicional é um método direto, no qual vamos "construindo" provas a partir dos axiomas, ou a partir das provas anteriores.

Existem outros métodos alternativos de demonstração, em particular alguns baseados na "desconstrução" das fórmulas, ou mais precisamente na *análise* das fórmulas (ao contrário do método axiomático, que se baseia na *análise*).

Esse método, conhecido como *método dos tablôs*[1] *analíticos*, tem suas raízes originariamente no estudo de Gerard Gentzen [3], discípulo de Hilbert, depois sistematizado nos trabalhos de Beth [1] e Hintikka [4]. Uma versão mais moderna é apresentada no livro de Smullyan [8], 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 elegante, é muito eficaz 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 próximo capítulo apresentaremos o método analítico em detalhes.

3.2 Sistemas Axiomáticos

A partir de uma linguagem fixada, a idéia do método axiomático é obter demonstrações de fórmulas a partir de um conjunto dado de premissas. Estas demonstrações são seqüências finitas de fórmulas, cada uma delas sendo

[1] Usamos 'tablô' e 'tablôs', respectivamente, como aportuguesamento de 'tableau' e 'tableaux'.

justificada pelas regras do sistema, terminando a seqüência com a fórmula que desejávamos demonstrar (caso tenhamos sucesso). As *justificativas* para colocar uma fórmula na seqüência são dadas na definição a seguir:

Definição 3.2.1. Uma *demonstração* num sistema formal é uma sequência finita $\gamma_1, \gamma_2, \cdots, \gamma_n$, em que γ_n é a fórmula que se quer demonstrar. A inserção de cada γ_i, com $1 \leq i \leq n$, na sequência é justificada se:

- γ_i é uma das premissas dadas, ou
- Se γ_i pertencer ao conjunto dos axiomas do sistema, ou
- Se γ_i for obtida a partir de fórmulas anteriores da seqüência por meio das *regras de inferência* do sistema.

□

Definição 3.2.2. Se Γ é um conjunto de premissas que permite demonstrar uma fórmula φ através dos passos acima, dizemos que φ *é consequência sintática de* Γ, denotado por $\Gamma \vdash \varphi$.

Pode ocorrer que o conjunto Γ de premissas seja infinito, o que parece um tanto absurdo, mas que em princípio não pode ser descartado. Veremos logo a seguir que esse caso é irrelevante, uma vez que sempre podemos "jogar fora" certas premissas de forma a manter apenas um conjunto finito de premissas (isto é exatemente o que diz a propriedade da Compacidade abaixo).

A partir da definição de "demonstração" dada acima, podemos obter algumas propriedades imediatas da relação de consequência sintática ou relação de forçamento sintático, cuja justificativa deixamos como exercício para o leitor:

Metateorema 3.2.3. *Num sistema axiomático valem as seguintes afirmações:*

1. $\varphi \vdash \varphi$ *(Reflexividade)*
2. *Se* $\vdash \varphi$ *então* $\Gamma \vdash \varphi$
3. *Se* $\varphi \in \Gamma$ *então* $\Gamma \vdash \varphi$
4. *Se* $\Gamma \vdash \varphi$ *e* $\Gamma \subseteq \Delta$ *então* $\Delta \vdash \varphi$ *(Monotonicidade)*
5. *Se* $\Gamma \vdash \varphi_i$ $(i = 1, \ldots, n)$ *e* $\varphi_1, \ldots, \varphi_n \vdash \psi$ *então* $\Gamma \vdash \psi$ *(Transitividade)*
6. *Se* $\Gamma \vdash \varphi$ *e* $\Delta, \varphi \vdash \psi$ *então* $\Gamma, \Delta \vdash \psi$

7. *Se* $\Gamma, \varphi \vdash \psi$ *e* $\Gamma \vdash \varphi$ *então* $\Gamma \vdash \psi$

8. $\Gamma \vdash \varphi$ *se e só se existe* $\Delta \subseteq \Gamma$ *(*Δ *finito) tal que* $\Delta \vdash \varphi$ *(Compacidade)*

Demonstração: Vamos demonstrar apenas alguns casos como ilustração, os demais são deixados para o leitor como exercício.

6 Dado que $\Delta, \varphi \vdash \psi$, pela Definição 3.2.2, sabemos que φ é uma das premissas usadas para se demonstrar ψ.

□

A justificativa das propriedades acima pode parecer complicada, mas é de fato imediata se se aplicar a definição de demonstração. A Compacidade, por exemplo, é consequência direta do fato de que um demonstração é uma sequência *finita* de passos: numa sequência finita não cabe um número infinito de premissas, e daí se obtém que, se $\Gamma \vdash \varphi$, existe Δ finito, $\Delta \subseteq \Gamma$, tal que $\Delta \vdash \varphi$. O conjunto Δ pode ser, por exemplo, o conjunto dos elementos de Γ utilizados, de fato, numa demonstração concreta (no sistema axiomático dado) de φ a partir de Γ. A importante metapropriedade da Compacidade será retomada na Seção 3.4.

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 $\leftrightarrow$ já definidas no capítulo anterior.

Definição 3.3.1. O sistema axiomático PC para a LPC consiste de um único axioma e quatro regras básicas 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) $\dfrac{\varphi \vee \psi \qquad \neg\varphi \vee \gamma}{\psi \vee \gamma}$

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

Cada vez que tivermos demonstrada uma relação de consequência sintática $\Gamma \vdash \varphi$ poderemos guardá-la como uma nova regra de derivação (ou inferência). Nesse caso, poderemos denotá-la por:

$$\frac{\Gamma}{\varphi}$$

Uma longa lista de metateoremas que demonstram novas regras de derivação é a seguinte; detalhamos as demonstrações de forma que possam ser seguidas passo a passo. O importante nao é neste momento se preocupar sobre como produzir tais demonstrações, mas somente entendê-las.

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

Demonstração:

1. φ [Hipótese]
2. $\varphi \vee \psi$ [Expansão em 1]
3. $\neg\varphi \vee \psi$ [Hipótese]
4. $\psi \vee \psi$ [Corte em 2 e 3]
5. ψ [Eliminação em 5]

□

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

Demonstração:

1. $\varphi \vee \psi$ [Hipótese]
2. $\neg\varphi \vee \varphi$ [Axioma]
3. $\psi \vee \varphi$ [Corte em 1 e 2]

□

Metateorema 3.3.4. $\gamma \vee (\varphi \vee \varphi) \vdash_{PC} \gamma \vee \varphi$. *(Cancelamento)*

Demonstração:

1. $\gamma \vee (\varphi \vee \varphi)$ [Hipótese]
2. $(\gamma \vee \varphi) \vee \varphi$ [Associatividade em 1]
3. $\varphi \vee (\gamma \vee \varphi)$ [Comutatividade em 2 (3.3.3)]
4. $[\varphi \vee (\gamma \vee \varphi)] \vee \gamma$ [Expansão em 3]
5. $\gamma \vee [\varphi \vee (\gamma \vee \varphi)]$ [Comutatividade em 4]
6. $(\gamma \vee \varphi) \vee (\gamma \vee \varphi)$ [Associatividade em 5]
7. $(\gamma \vee \varphi)$ [Eliminação em 6]

□

Metateorema 3.3.5. $(\varphi \vee \psi) \vee \gamma \vdash_{PC} \varphi \vee (\psi \vee \gamma)$. *(Associatividade)*

Demonstração:

1. $(\varphi \vee \psi) \vee \gamma$ [Hipótese]
2. $\gamma \vee (\varphi \vee \psi)$ [Comutatividade em 1 (3.3.3)]
3. $(\gamma \vee \varphi) \vee \psi$ [Associatividade em 2]
4. $\psi \vee (\gamma \vee \varphi)$ [Comutatividade em 3 (3.3.3)]
5. $(\psi \vee \gamma) \vee \varphi$ [Associatividade em 4]
6. $\varphi \vee (\psi \vee \gamma)$ [Comutatividade em 5]

□

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

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.) □

Metateorema 3.3.7. $\varphi \vee (\psi \vee \gamma) \vdash_{PC} \psi \vee (\gamma \vee \varphi)$. *(Permutação)*

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) □

Metateorema 3.3.8. $\varphi \vee (\psi \vee \gamma) \vdash_{PC} \gamma \vee (\varphi \vee \psi)$. *(Permutação, segunda forma)*

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) □

Metateorema 3.3.9. $\vdash_{PC} \neg(\varphi \vee \psi) \vee (\psi \vee \varphi)$. *(Axioma + Comutatividade)*

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) □

Metateorema 3.3.10. $\varphi \vee (\psi \vee \gamma) \vdash_{PC} \varphi \vee (\gamma \vee \psi)$. *(Permutação, terceira forma)*

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)) □

Metateorema 3.3.11. *(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) □

Metateorema 3.3.12. $\varphi, \neg\varphi \vdash_{PC} \psi$. *(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.) □

A partir das inferências acima, podemos resumir algumas regras de inferência bem conhecidas:

- (Modus Tollendo Ponens, primeira forma) $\dfrac{\varphi, \neg\varphi \vee \psi}{\psi}$
- (Modus Tollendo Ponens, segunda forma) $\dfrac{\neg\varphi, \varphi \vee \psi}{\psi}$

  a partir de (Modus Tollendo Ponens, primeira forma), lembrando que $\neg\neg\varphi \equiv \varphi$

- (Modus Ponendo Ponens) $\dfrac{\varphi, \varphi \to \psi}{\psi}$

  a partir de (Modus Tollendo Ponens), lembrando que $\neg\varphi \vee \psi \equiv \varphi \to \psi$

- (Silogismo Disjuntivo) $\dfrac{\varphi \to \alpha, \psi \to \beta, \varphi \vee \psi}{\alpha \vee \beta}$

  a partir dos anteriores (exercício).

Uma propriedade que simplifica bastante o manuseio dos teoremas e demonstrações é o Metateorema da Dedução, na forma sintática, que permite transformar a prova de uma dedução na prova de uma implicação. (Lembramos que a implicação $\varphi \to \psi$ 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, também foi demonstrado no capítulo anterior, no Metateorema 2.2.28(9).

Metateorema 3.3.13. (Metateorema da Dedução, forma sintática) $\Gamma, \varphi \vdash_{PC} \psi$ *see* $\Gamma \vdash_{PC} \neg\varphi \vee \psi$.

Demonstração:
($\Longleftarrow$)

1.	$\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \psi$	[Hipótese]
2.	$\Gamma, \varphi \vdash_{\mathsf{PC}} \varphi$	[Reflexividade e Monotonicidade de $\vdash_{\mathsf{PC}}$]
3.	$\Gamma, \varphi \vdash_{\mathsf{PC}} \varphi \vee \psi$	[Expansão em 2]
4.	$\Gamma, \varphi \vdash_{\mathsf{PC}} \neg\varphi \vee \psi$	[Monotonicidade em 1]
5.	$\Gamma, \varphi \vdash_{\mathsf{PC}} \psi \vee \psi$	[Corte em 3 e 4]
6.	$\Gamma, \varphi \vdash_{\mathsf{PC}} \psi$	[Eliminação em 5]

($\Longrightarrow$) Suponha que $\Gamma, \varphi \vdash_{\mathsf{PC}} \psi$, isso significa que existe uma sequência finita $\gamma_1, \gamma_2, \cdots, \gamma_n$, onde $\gamma_n = \psi$. Vamos mostrar, por indução no comprimento da prova, que $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \psi$.

1. Para $n = 1$.
   Nesse caso temos que a sequência $\gamma_1, \gamma_2, \cdots, \gamma_n$ é composta de um único elemento, daí pela definição de prova temos que γ_1 é ou um axioma do sistema ou uma premissa de Γ.

   - Se $\gamma_1 = \psi$ é um axioma. A partir dessa sequência (ou prova), podemos obter o seguinte:

1.	$\vdash_{\mathsf{PC}} \psi$	[Axioma]
2.	$\vdash_{\mathsf{PC}} \psi \vee \neg\varphi$	[Expansão em 1]
3.	$\vdash_{\mathsf{PC}} \neg\varphi \vee \psi$	[Comutatividade em 2]
4.	$\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \psi$	[Metateorema 3.2.3 (2), em 3]

- Se $\gamma_1 = \psi$ é uma premissa de $\Gamma \cup \{\varphi\}$. Nesse caso temos que $\psi = \varphi$ ou $\psi \in \Gamma$.
  - Se $\psi \in \Gamma$ então $\Gamma \vdash_{\mathsf{PC}} \psi$ (Metateorema 3.2.3 (3)). Daí, pelas regras de Expansão e Comutatividade segue que $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \psi$
  - Se $\psi = \varphi$, pela reflexividade temos que $\varphi \vdash_{\mathsf{PC}} \varphi$. Daí, por Expansão e Comutatividade, segue que $\varphi \vdash_{\mathsf{PC}} \neg\varphi \vee \varphi$. Mas $\neg\varphi \vee \varphi$ é axioma. Logo, $\vdash_{\mathsf{PC}} \neg\varphi \vee \varphi$. Portanto, pela monotonicidade, segue que $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \varphi$.

2. Para $n \leq k$ suponha, por hipótese de indução, que o resultado seja válido.

3. Para $n = k + 1$
   Nos casos em que ψ é axioma ou $\psi \in \Gamma \cup \{\varphi\}$ o argumento é análogo ao caso $n = 1$. Resta verificar os casos em que ψ foi obtida pela aplicação de uma das regras de PC.

   - Se ψ foi obtida por Expansão.
     Isso significa que existe uma linha i na prova, com $1 \leq i \leq k$, tal que $\psi = \gamma_i \vee \alpha$, para alguma fórmula α. Como γ_i tem complexidade menor que ψ então, pela hipótese de indução, temos que $\Gamma, \varphi \vdash_{\mathsf{PC}} \gamma_i$ sse $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \gamma_i$. Daí, por expansão, temos que $\Gamma \vdash_{\mathsf{PC}} (\neg\varphi \vee \gamma_i) \vee \alpha$, e pela associatividade temos que $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee (\gamma_i \vee \alpha)$, isto é, $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \psi$.
   - Se ψ foi obtida por Eliminação.
     Isso significa que existe uma linha i na prova, com $1 \leq i \leq k$, tal que $\gamma_i = \psi \vee \psi$. Como γ_i tem complexidade menor que ψ então, pela hipótese de indução, temos que $\Gamma, \varphi \vdash_{\mathsf{PC}} \psi \vee \psi$ sse $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee (\psi \vee \psi)$. Daí, por cancelamento, temos que $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \psi$.
   - Se ψ foi obtida pela Associatividade.
     Exercício.
   - Se ψ foi obtida pelo Corte.
     Exercício.

□

Metateorema 3.3.14. (Metateorema da Dedução, forma sintática) $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\} \vdash_{\mathsf{PC}} \psi$ *se, e só se,*

$$\Gamma \vdash_{\mathsf{PC}} \varphi_1 \to (\varphi_2 \to (\ldots \to (\varphi_n \to \psi)\ldots)).$$

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

Metateorema 3.3.15.
(a) $\Gamma, \varphi \vdash_{PC} \gamma$ *e* $\Gamma, \psi \vdash_{PC} \gamma$ *implica* $\Gamma, \varphi \vee \psi \vdash_{PC} \gamma$.
(b) $\Gamma, \varphi \vdash_{PC} \gamma$ *e* $\Gamma, \neg\varphi \vdash_{PC} \gamma$ *implica* $\Gamma \vdash_{PC} \gamma$. *(Prova por casos)*

Demonstração:

(a) Segue imediatamente pelo Metateorema da Dedução 3.3.13 e pela Junção.

(b) A partir das hipóteses, usando o Metateorema da Dedução 3.3.13 e a Junção, segue que $\Gamma \vdash_{\mathsf{PC}} \neg(\neg\varphi \vee \varphi) \vee \gamma$. Dado que $\neg\varphi \vee \varphi$ é axioma de PC segue que $\Gamma \vdash_{\mathsf{PC}} \neg\varphi \vee \varphi$. Daí, pelas regras de Expansão, Corte e Eliminação segue que $\Gamma \vdash_{\mathsf{PC}} \gamma$.

□

3.4 Completude e Compacidade

Nesta seção discutimos dois metateoremas importantíssimos da lógica proposicional clássica: o Metateorema da Completude e o Metateorema da Compacidade. Como veremos a seguir, ambos os resultados estão interrelacionados.

O Teorema da Completude para PC garante que qualquer consequência semântica, que pode ser obtida por meio de tabelas-verdade, coincide com uma demonstração sintática. Este resultado é ao mesmo tempo muito útil, e muito significativo. É útil porque, na Lógica Proposicional Clássica, é em geral mais fácil manipular com a parte semântica; e é extremamente significativo porque deixa claro que tudo o que se consegue com demonstrações em PC consegue-se igualmente com tabelas-verdade; mostra ainda mais que o axioma e regras que propusemos são totalmente suficientes para demonstrar *qualquer* tautologia em PC.

Metateorema 3.4.1. (Teorema de Correção para PC) *Seja* φ *uma fórmula em* Prop *e* Γ *um conjunto de fórmulas em* Prop. *Então* $\Gamma \vdash_{PC} \varphi$ *implica* $\Gamma \models \varphi$.

Demonstração:

- Para o caso em que $\Gamma = \emptyset$ tem-se a chamada *correção fraca*. Nesse caso basta verificar por meio da tabela-verdade que o axioma de PC $\neg\varphi \vee \varphi$ é uma tautologia.

- Para o caso em que $\Gamma \neq \emptyset$ tem-se a chamada *correção forte*. Nesse caso, a prova deve levar em consideração o uso das regras de inferência de PC na dedução de φ.

  Se $\Gamma \vdash_{\mathsf{PC}} \varphi$, isso significa que existe uma sequência $\gamma_1, \gamma_2, \cdots, \gamma_n$ tal que $\gamma_n = \varphi$ e cada γ_i, com $1 \leq i \leq n$, é ou uma axioma de PC, ou um elemento de Γ ou foi obtida por meio de uma das regras de inferência de PC. A prova é feita por indução no comprimento da prova.

  - Para $n = 1$
    Nesse caso ou φ é um axioma de PC ou $\varphi \in \Gamma$.
    1. Se φ é um axioma, pela correção fraca, segue que $\models_{\mathsf{PC}} \varphi$. Pela monotonicidade segue que $\Gamma \models_{\mathsf{PC}} \varphi$.
    2. Se $\varphi \in \Gamma$, então $\Gamma \models \varphi$ (Metateorema 2.2.28 (3)).

  - Para $n \leq k$ é suposto, por hipótese de indução, que o resultado seja válido.
  - Para $n = k + 1$
    Nesse caso ou φ é um axioma de PC ou $\varphi \in \Gamma$, ou φ foi obtida pela aplicação de uma das regras de inferência de PC.
    1. Se φ é um axioma ou $\varphi \in \Gamma$, análogo ao caso $n = 1$.
    2. Se φ foi obtida por Expansão.
       Isso significa que existe uma linha i na prova, com $1 \leq i \leq k$, tal que $\varphi = \gamma_i \vee \alpha$ foi obtida pela aplicação da regra de expansçao na linha i. Pela hipótese de indução sabemos que se $\Gamma \vdash_{\mathsf{PC}} \gamma_i$, então $\Gamma \models_{\mathsf{PC}} \gamma_i$, para $1 \leq i \leq k$, ou seja, γ_i é uma consequência semântica de Γ, mas isso significa que para toda valoração v, se $v(\Gamma) = 1$, então $v(\gamma_i) = 1$. Agora, se $v(\gamma_i) = 1$ então $v(\gamma_i \vee \alpha) = 1$. Portanto, $\Gamma \models \gamma_i \vee \alpha$, ou seja, $\Gamma \models \varphi$.
    3. Se φ foi obtida por Eliminação.
       Isso significa que existe uma linha i na prova, com $1 \leq i \leq k$, tal que $\gamma_i = \varphi \vee \varphi$. Pela hipótese de indução sabemos $\Gamma \models_{\mathsf{PC}} \gamma_i$, para $1 \leq i \leq k$, ou seja, $\varphi \vee \varphi$ é uma consequência semântica de Γ, mas isso significa que para toda valoração v, se $v(\Gamma) = 1$, então $v(\varphi \vee \varphi) = 1$. Agora, $v(\varphi \vee \varphi) = 1$ sse $v(\varphi) = 1$. Portanto, $\Gamma \models \varphi$.
    4. Se φ foi obtida por Associatividade.
       Exercício.

5. Se φ foi obtida por Corte.
   Exercício.

□

Para a obtenção da prova da completude fraca basta mostrar, primeiramente, o seguinte resultado.

Metateorema 3.4.2. *Seja φ uma fórmula em Prop e $Var(\varphi) = \{q_1, \cdots, q_n\}$. Seja v uma valoração qualquer. Para $1 \leq i \leq n$ defina:*

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

Seja $\Delta_\varphi = \{q_1^*, \cdots, q_n^*\}$. Então:

(a) *Se $v(\varphi) = 1$, então $\Delta_\varphi \vdash_{PC} \varphi$.*

(b) *Se $v(\varphi) = 0$, então $\Delta_\varphi \vdash_{PC} \neg\varphi$.*

Demonstração: Por indução na complexidade de φ

- Se $\varphi \in \mathcal{V}$.
  Nesse caso temos que $\varphi = q_i$, para algum $q_i \in \mathcal{V}$.

  (a) Se $v(\varphi) = 1$, então $v(q_i) = 1$, daí $q_i^* = q_i$. Pelo Metateorema 3.2.3 (1) temos que $q_i \vdash_{\mathsf{PC}} q_i$, isto é, $\Delta_\varphi \vdash_{\mathsf{PC}} \varphi$.

  (b) Se $v(\varphi) = 0$, então $v(q_i) = 0$, daí $q_i^* = \neg q_i$. Pelo Metateorema 3.2.3 (1) temos que $\neg q_i \vdash_{\mathsf{PC}} \neg q_i$, isto é, $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\varphi$.

- Vamos supor, por hipótese de indução, que o resultado seja válido para fórmulas de complexidade menor que φ.

- Se $\varphi = \neg\psi$.

  (a) Se $v(\varphi) = 1$, então $v(\psi) = 0$. Pela hipótese de indução temos que $\Delta_\psi \vdash_{\mathsf{PC}} \neg\psi$. Como $\varphi = \neg\psi$, então $\Delta_\psi = \Delta_\varphi$. Portanto, $\Delta_\varphi \vdash_{\mathsf{PC}} \varphi$.

  (b) Se $v(\varphi) = 0$, então $v(\psi) = 1$. Pela hipótese de indução temos que $\Delta_\psi \vdash_{\mathsf{PC}} \psi$. Mas, $\psi \vdash_{\mathsf{PC}} \neg\neg\psi$ (Lei da dupla negação), daí, pelo Metateorema 3.2.3 (5), segue que $\Delta_\psi \vdash_{\mathsf{PC}} \neg\neg\neg\psi$. Portanto, $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\varphi$.

- Se $\varphi = \psi_1 \vee \psi_2$.

(a) Se $v(\varphi) = 1$, então $v(\psi_1) = 1$ ou $v(\psi_2) = 1$. Vamos considerar o caso em que $v(\psi_1) = 1$, o outro caso é análogo. Como ψ_1 tem complexidade menos que φ então, pela hipótese de indução, temos que $\Delta_{\psi_1} \vdash_{\mathsf{PC}} \psi_1$ daí, pela expansão, $\Delta_{\psi_1} \vdash_{\mathsf{PC}} \psi_1 \vee \psi_2$. Como $\Delta_{\psi_1} \subseteq \Delta\varphi$ então, pelo Metateorema 3.2.3 (4), temos que $\Delta_\varphi \vdash_{\mathsf{PC}} \psi_1 \vee \psi_2$, isto é, $\Delta_\varphi \vdash_{\mathsf{PC}} \varphi$.

(b) Se $v(\varphi) = 0$, então $v(\psi_1) = 0$ e $v(\psi_2) = 0$. Como ψ_1 e ψ_2 têm complexidade menor que φ então, pela hipótese de indução, temos que $\Delta_{\psi_1} \vdash_{\mathsf{PC}} \neg\psi_1$ e $\Delta_{\psi_2} \vdash_{\mathsf{PC}} \neg\psi_2$. Como $\Delta_{\psi_1} \subseteq \Delta\varphi$ e $\Delta_{\psi_2} \subseteq \Delta\varphi$ então, pelo Metateorema 3.2.3 (4), temos que $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\psi_1$ e $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\psi_2$. Aplicando a regra de expansão temos que $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\psi_1 \vee \neg(\psi_1 \vee \psi_2)$ e $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\psi_2 \vee \neg(\psi_1 \vee \psi_2)$, e aplicando a junção temos que $\Delta_\varphi \vdash_{\mathsf{PC}} \neg(\psi_1 \vee \psi_2) \vee \neg(\psi_1 \vee \psi_2)$. Portanto, por eliminação, $\Delta_\varphi \vdash_{\mathsf{PC}} \neg(\psi_1 \vee \psi_2)$, isto é, $\Delta_\varphi \vdash_{\mathsf{PC}} \neg\varphi$.

□

Como uma consequência do Teorema 3.4.2 tem-se a completude fraca para PC.

Metateorema 3.4.3. (Teorema da Completude para PC) *Seja φ uma fórmula em Prop e Γ um conjunto de fórmulas em Prop. Então $\Gamma \models \varphi$ se e somente se $\Gamma \vdash_{PC} \varphi$.*

Demonstração: Se $\models \varphi$, isto é, se φ é uma tautologia, então $v(\varphi) = 1$ para toda valoração v. Sejam $Var(\varphi) = \{q_1, \cdots, q_n\}$ e v_1 e v_2 valorações tais que:

$$v_1(q_i) = 1 \quad \text{para } 1 \leq i \leq n$$

$$v_2(q_i) = \begin{cases} 1 & \text{para } 1 \leq i \leq n-1; \\ 0 & \text{para } i = n \end{cases}$$

Com base nessas valorações construímos os conjuntos Δ^1_φ e Δ^2_φ de acordo com o Lema 3.4.2:

$$\begin{aligned} \Delta^1_\varphi &= \{q_1, \cdots, q_{n-1}, q_n\} \\ \Delta^2_\varphi &= \{q_1, \cdots, q_{n-1}, \neg q_n\} \end{aligned}$$

Como $v(\varphi) = 1$ para toda valoração v então, pelo Lema 3.4.2 (a) temos:

$$\Delta^1_\varphi \vdash_{\mathsf{PC}} \varphi$$
$$\Delta^2_\varphi \vdash_{\mathsf{PC}} \varphi.$$

Aplicando o Metateorema da Dedução 3.3.13 temos:

$$\{q_1, \cdots, q_{n-1}\} \vdash_{\mathsf{PC}} \neg q_n \vee \varphi$$
$$\{q_1, \cdots, q_{n-1}\} \vdash_{\mathsf{PC}} \neg\neg q_n \vee \varphi.$$

Aplicando a regra do corte e a regra de eliminação segue que:

$$\{q_1, \cdots, q_{n-1}\} \vdash_{\mathsf{PC}} \varphi$$

Repetindo esse mesmo raciocínio $n-1$-vezes, todas as premissas do conjunto $\{q_1, \cdots, q_{n-1}\}$ são eliminadas, ou seja, $\vdash_{\mathsf{PC}} \varphi$. [2]

□

Como corolário do Teorema da Completude, 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.

Corolário 3.4.4. (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, pela completude forte, temos que $\Gamma \vdash_{\mathsf{PC}} \varphi$ daí, pela compacidade sintática, temos que existe um conjunto finito $\Gamma_0 \subseteq \Gamma$ tal que $\Gamma_0 \vdash_{\mathsf{PC}} \varphi$. Portanto, pela completude forte, segue que $\Gamma_0 \models \varphi$.

□

Observe que uma versão sintática deste resultado aparece no Metateorema 3.2.3(8).

3.5 Outras Axiomáticas

A axiomática para PC utilizada na seção anterior tem apenas um axioma e diversas regras de inferência. Existem muitas outras axiomáticas, considerando a mesma linguagem (ou outras 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 [2].

[2] Note que ao usar n vezes o mesmo raciocínio para eliminar as n variáveis de φ, na verdade estamos usando um método que necessita 2^n valorações. Se conseguíssemos menos que 2^n valorações estaríamos resolvendo o famoso problema $P \stackrel{?}{=} NP$!

A seguinte axiomática foi popularizada no conhecido livro de Mendelson [6], embora tenha sido introduzida por D. Hilbert e P. Bernays, e é sem dúvida uma das axiomáticas da LPC mais divulgadas na literatura:

Definição 3.5.1. Seja Prop_0 a linguagem gerada pela assinatura $C_0 = \{\neg, \rightarrow\}$ do Exemplo 2.1.3. O sistema axiomático M para a LPC escrito na linguagem Prop_0 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 axioma e regra de M pode ser demonstrado 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_0. Portanto, M também satisfaz, tanto quanto PC, o Teorema da Dedução e o Teorema da Completude.

3.6 Exercícios

1. Considere as seguintes identidades:

   - $\alpha \rightarrow \beta = \neg\alpha \vee \beta$
   - $\alpha \wedge \beta = \neg(\neg\alpha \vee \neg\beta)$

   Com base nas identidades acima, traduza as fórmulas abaixo para fórmulas da assinatura $C = \{\neg, \vee\}$.

   (a) $\varphi \rightarrow (\psi \rightarrow \varphi)$;
   (b) $\varphi \rightarrow (\psi \rightarrow \gamma) \rightarrow [(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \gamma)]$
   (c) $\neg\varphi \rightarrow \neg\psi[(\neg\varphi \rightarrow \psi) \rightarrow \varphi]$
   (d) $[(\varphi \wedge \neg\gamma) \rightarrow \delta] \rightarrow \{(\psi \wedge \neg\delta) \rightarrow [(\varphi \wedge \neg\gamma) \rightarrow \delta]\}$;

(e) $\neg\varphi \to (\psi \vee \neg\varphi)$

(f) $(\varphi \wedge \psi) \to \psi$

(g) $\neg\varphi \to [\neg\psi \to (\neg\varphi \wedge \neg\psi)]$

(h) $[(\varphi \to \psi) \to (\gamma \to \psi)] \to [(\varphi \vee \gamma) \to \psi]$

(i) $[(\varphi \to \psi) \to (\varphi \to \gamma)] \to [\varphi \to (\psi \wedge \gamma)]$

(j) $(\varphi \to \psi) \to [(\varphi \to \psi) \to \psi] \to \psi$

2. Mostre, sem fazer uso do Metateorema da Dedução, que as traduções dadas no exercício anterior são teoremas na axiomática para PC escrita na assinatura $\{\neg, \vee\}$.

3. Mostre, sem fazendo uso do Metateorema da Dedução, que as traduções dadas no exercício anterior são teoremas na axiomática para PC escrita na assinatura $\{\neg, \vee\}$.

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) $\vdash \varphi \to \varphi$

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

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

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

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

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

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

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

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

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

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

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

- **Regras de introdução de conectivos**
  (a) Se $\Gamma, \varphi \vdash \psi$ então $\Gamma \vdash \varphi \to \psi$
  (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) $\varphi, \varphi \to \psi \vdash \psi$
  (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) $\neg\neg\varphi \vdash \varphi$

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", analisado no Capítulo 3.

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, daí 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. [4]) e Smullyan (cf. [8]). 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; isto é, usaremos apenas os conectivos $\neg$ e $\vee$, sendo que os outros conectivos serão expressados através das abreviações usuais, como foi feito no capítulo anterior .

Definição 4.1.1. Introduzimos as seguintes regras de análise:

$$(R_{\vee}) \quad \frac{\varphi \vee \psi}{\varphi \mid \psi}$$

$$(R_{\neg\vee}) \quad \frac{\neg(\varphi \vee \psi)}{\neg\varphi, \neg\psi}$$

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

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 $R_{\neg\neg}$).

[1] Lembre que decidimos adotar os neologismos "tablô" e "tablôs" para os termos franceses *tableau* e *tableaux*, respectivamente.

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 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 é 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

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

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

(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

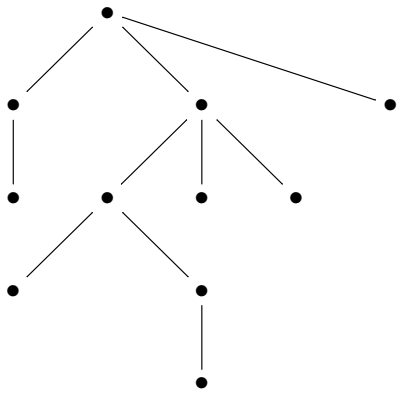
$$x_1\mathsf{R}x_2\mathsf{R}\cdots\mathsf{R}x_n\mathsf{R}x_1.$$

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*.

Uma árvore é representada da maneira seguinte:



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

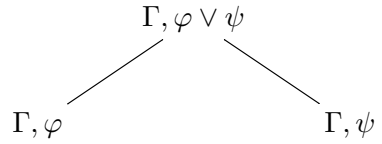
1. Uma seqüência finita $x_1 \cdots x_n$ em $|\mathcal{A}|$ é um *ramo finito* de $\mathcal{A}$ se $x_1 = r$, x_n é terminal e $x_i\mathsf{R}x_{i+1}$ para todo $1 \leq i \leq n-1$.

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 $i \geq 1$.

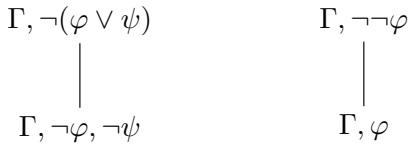
3. Um *ramo* de $\mathcal{A}$ é um ramo finito ou infinito de $\mathcal{A}$.

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

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,



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



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:

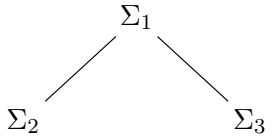
   (a) Se aplicamos $(R_{\neg\vee})$ ou $(R_{\neg\neg})$ num nó terminal Σ' de $\mathcal{A}_n$ obtendo um sucessor Δ de Σ' em $\mathcal{A}_{n+1}$, então definimos C_{n+1} como sendo a configuração obtida de C_n substituindo Σ' por Δ. Note que os nós terminais de $\mathcal{A}_{n+1}$ são exatamente os elementos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

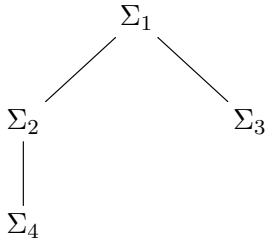
Por exemplo, considere o seguinte tablô $\mathsf{C}_1\mathsf{C}_2\mathsf{C}_3$ e a seqüência de árvores correspondente:

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

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



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



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 $\mathcal{A}_n$.

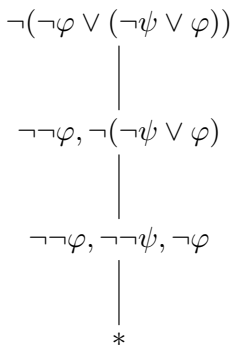
Definição 4.1.5. Dizemos que Γ *deriva* φ *analiticamente* se existe um tablô fechado para o conjunto $\Gamma, \neg\varphi$; denotamos tal fato por $\Gamma \vdash_T \varphi$. Dizemos que φ é um *teorema obtido analiticamente* se existe um tablô fechado para $\{\neg\varphi\}$ (isto é, se φ se deriva do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

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 $\neg\varphi$.

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

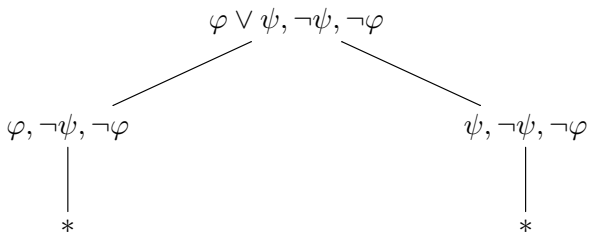
Exemplo 4.1.6. $\vdash_T \varphi \to (\psi \to \varphi)$. Iniciamos uma árvore com $\neg(\varphi \to$

$(\psi \to \varphi))$ e mostramos que se produz um tablô fechado.



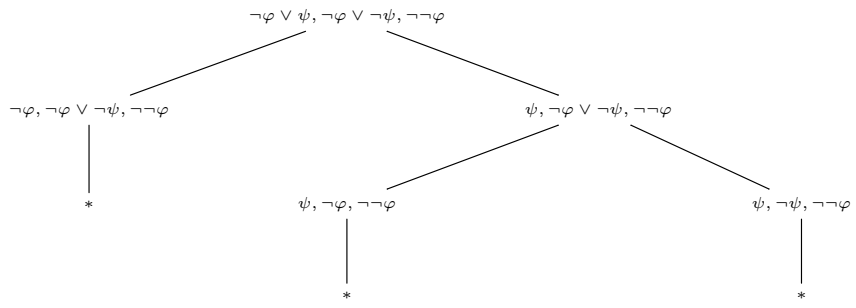
(o ramo fecha em virtude da ocorrência de $\neg\neg\varphi$ e $\neg\varphi$. △

Exemplo 4.1.7. $\varphi \vee \psi, \neg\psi \vdash_T \varphi$.



△

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



△

É relativamente simples provar que relação de conseqüência $\vdash_T$ satisfaz as propriedades (1)-(4) e (8) do Metateorema 3.2.3. Por outro lado, a

transitividade de $\vdash_T$ é mais complicada de ser provada: 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.

Com esta propriedade, pode-se demonstrar a completude do método analítico e sua equivalência com o método hilbertiano. A demonstração deste resultado não é simples, e não será feita aqui.

Finalizamos esta subseção com o estudo de regras derivadas.

Definição 4.1.9. Seja $\Gamma \subseteq \mathsf{Prop}$. Dizemos que Γ é *T-inconsistente* se existe um tablô fechado para Γ.

Definição 4.1.10. 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:

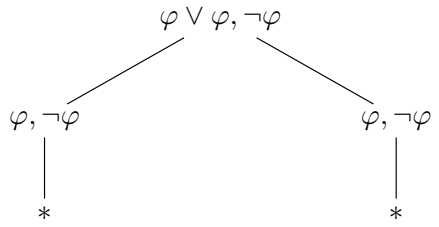
1. φ_1 e φ_2 (não necessariamente distintas) são subfórmulas de complexidade estritamente menor do que a complexidade de φ, e
2. $\Gamma \cup \{\varphi, \neg\varphi_1\}$ e $\Gamma \cup \{\varphi, \neg\varphi_2\}$ são T-inconsistentes.

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 4.1.11. As seguintes são regras derivadas:

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



2. $\dfrac{\varphi \to \psi}{\neg\varphi | \psi}$ porque $\{\neg\varphi \vee \psi, \neg\neg\varphi, \neg\psi\}$ é T-inconsistente.

3. $\dfrac{\neg(\varphi \to \psi)}{\varphi, \neg\psi}$ (exercício).

4. $\dfrac{\varphi \wedge \psi}{\varphi, \psi}$ (exercício).

5. $\dfrac{\neg(\varphi \wedge \psi)}{\neg\varphi | \neg\psi}$ (exercício).

△

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:

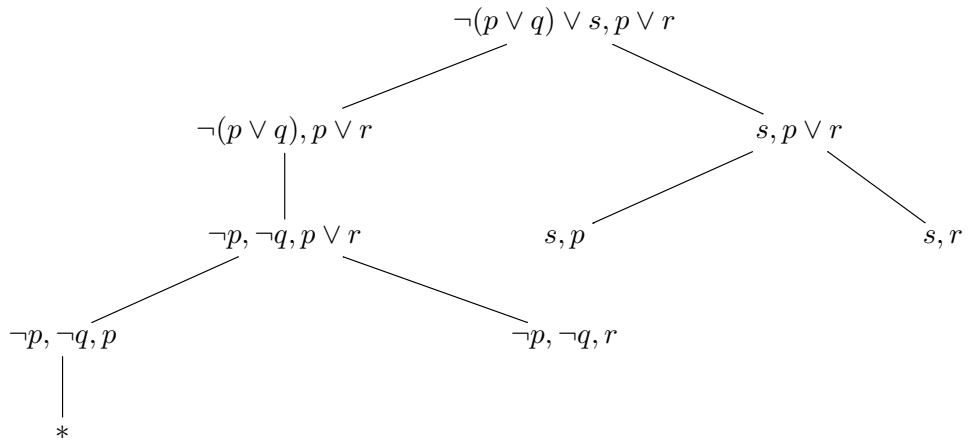
Para todo $\Gamma \cup \{\varphi\}$, $\Gamma \vdash_T \varphi$ *see* $\Gamma \vdash_{\mathsf{PC}} \varphi$.

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, $\Gamma \nvdash_T \varphi$. Um dos grandes problemas da Teoria da Computação, como já discutimos, é saber se existem ou não métodos mais eficazes que estes dois métodos descritos acima. Tal questão está ligada ao problema $P \stackrel{?}{=} NP$, um dos mais difíceis problemas da computação teórica.

Finalizamos esta seção observando que o método de tablôs nos proporciona 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.

Exemplo 4.1.12. 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. $v_3(s) = 1$, $v_3(r) = 1$.

[4] Veremos no Capítulo 6 que esta propriedade não é mais válida nos tablôs para a lógica de primeira ordem (dado que esta lógica é indecidível).

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, pode-se obter seguinte resultado, cuja prova não faremos:

Corolário 4.1.13. *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. [3]), 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. [5]).

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

$\perp$) 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 ([7]), que transformou-se num clássico da área.

Definição 4.2.1. Considere a assinatura C^5 tal que $C^5_0 = \{\perp\}$, $C^5_2 = \{\vee, \wedge, \rightarrow\}$ 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 $\varphi \rightarrow \perp$):

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

$$(\vee I1)\ \frac{\varphi}{\varphi \vee \psi} \qquad (\vee I2)\ \frac{\psi}{\varphi \vee \psi} \qquad (\vee E)\ \frac{\varphi \vee \psi \qquad \overset{[\varphi]}{\gamma} \qquad \overset{[\psi]}{\gamma}}{\gamma}$$

$$(\rightarrow I)\ \frac{\overset{[\varphi]}{\psi}}{\varphi \rightarrow \psi} \qquad (\rightarrow E)\ \frac{\varphi \qquad \varphi \rightarrow \psi}{\psi}$$

$$(\perp 1)\ \frac{\perp}{\varphi} \qquad (\perp 2)\ \frac{\overset{[\neg\varphi]}{\perp}}{\varphi}$$

Temos as seguintes restrições nas regras de $\perp$: na regras $(\perp 1)$ e $(\perp 2)$ a fórmula φ é diferente de $\perp$ e, na regra $(\perp 2)$, a fórmula φ não é da forma $\neg\gamma$.

As restrições nas regras de $\perp$ 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 $(\perp 2)$ 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

$$\dfrac{\varphi \rightarrow \gamma \quad \varphi}{\gamma}$$

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:

$$\dfrac{\dfrac{\varphi \rightarrow \gamma \quad [\varphi]}{\gamma} \qquad \dfrac{\psi \rightarrow \gamma \quad [\psi]}{\gamma} \qquad \varphi \vee \psi}{\gamma}$$

Isto significa que γ é derivável em DN a partir de $\{\varphi \to \gamma, \psi \to \gamma, \varphi \vee \psi\}$. Finalmente, aplicando a regra $(\to I)$ podemos descarregar a hipótese $\varphi \vee \psi$, obtendo uma derivação de $(\varphi \vee \psi) \to \gamma$ em DN a partir de $\{\varphi \to \gamma, \psi \to \gamma\}$.

$$\dfrac{\dfrac{\dfrac{\varphi \to \gamma \quad [\varphi]^1}{\gamma} \qquad \dfrac{\psi \to \gamma \quad [\psi]^1}{\gamma} \qquad [\varphi \vee \psi]^2}{\gamma}\,1}{(\varphi \vee \psi) \to \gamma}\,2$$

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 $\gamma \to (\varphi \wedge \psi)$ é deduzido em DN a partir da hipótese $(\gamma \to \varphi) \wedge (\gamma \to \psi)$. A seguinte derivação em DN prova esse fato:

$$\dfrac{\dfrac{[\gamma]^1 \quad \dfrac{(\gamma \to \varphi) \wedge (\gamma \to \psi)}{\gamma \to \varphi}}{\varphi} \qquad \dfrac{[\gamma]^1 \quad \dfrac{(\gamma \to \varphi) \wedge (\gamma \to \psi)}{\gamma \to \psi}}{\psi}}{\dfrac{\varphi \wedge \psi}{\gamma \to (\varphi \wedge \psi)}\,1}$$

(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 $\neg\varphi \to \bot$):

$$\dfrac{\dfrac{\neg\neg\varphi \quad [\neg\varphi]^1}{\bot}}{\varphi}\,1 \qquad\qquad \dfrac{\dfrac{\varphi \quad [\neg\varphi]^1}{\bot}}{\neg\neg\varphi}\,1$$

(4) Provaremos que em DN a fórmula $\neg\varphi \vee \varphi$ é um teorema. Considere a seguinte derivação (lembrando que $\neg\varphi$ é $\varphi \to \bot$):

$$\dfrac{\dfrac{\dfrac{\dfrac{[\varphi]^1}{\neg\varphi \vee \varphi} \quad [\neg(\neg\varphi \vee \varphi)]^2}{\bot}}{\neg\varphi}\,1}{\neg\varphi \vee \varphi} \qquad [\neg(\neg\varphi \vee \varphi)]^2}{\dfrac{\bot}{\neg\varphi \vee \varphi}\,2}$$

Na derivação acima, o número 1 indica uma aplicação da regra $(\to I)$, enquanto que o número 2 indica uma aplicação da regra $(\bot 2)$. $\triangle$

O sistema DN é correto e completo para a semântica da lógica proposicional clássica. Isto é:

Metateorema 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.

Nesta seção utilizaremos a assinatura $C_3 = \{\top, \bot, \neg, \vee, \wedge, \rightarrow\}$. Nesta assinatura, a constante $\top$ representa o valor de verdade 1, enquanto que a constante $\bot$ representa, como sempre, o valor de verdade 0.

Definição 4.3.1. Um *sequente* é um par $\langle \Gamma, \Delta \rangle$ de conjuntos finitos de fórmulas na assinatura C_3. Um sequente $\langle \Gamma, \Delta \rangle$ será denotado por $\Gamma \Longrightarrow \Delta$.

Definição 4.3.2. O cálculo SQ de sequentes para a lógica proposicional clássica é dado pelas seguintes regras:

$$(Ax)\ \frac{}{\Gamma, \varphi \Longrightarrow \Delta, \varphi} \qquad (Corte)\ \frac{\Gamma \Longrightarrow \Delta, \varphi \quad \Gamma, \varphi \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta}$$

$$(\wedge E)\ \frac{\Gamma, \varphi, \psi \Longrightarrow \Delta}{\Gamma, \varphi \wedge \psi \Longrightarrow \Delta} \qquad (\wedge D)\ \frac{\Gamma \Longrightarrow \Delta, \varphi \quad \Gamma \Longrightarrow \Delta, \psi}{\Gamma \Longrightarrow \Delta, \varphi \wedge \psi}$$

$$(\rightarrow E)\ \frac{\Gamma \Longrightarrow \Delta, \varphi \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \rightarrow \psi \Longrightarrow \Delta} \qquad (\rightarrow D)\ \frac{\Gamma, \varphi \Longrightarrow \Delta, \psi}{\Gamma \Longrightarrow \Delta, \varphi \rightarrow \psi}$$

$$(\vee E)\ \frac{\Gamma, \varphi \Longrightarrow \Delta \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \vee \psi \Longrightarrow \Delta} \qquad (\vee D)\ \frac{\Gamma \Longrightarrow \Delta, \varphi, \psi}{\Gamma \Longrightarrow \Delta, \varphi \vee \psi}$$

$$(\neg E)\ \frac{\Gamma \Longrightarrow \Delta, \varphi}{\Gamma, \neg\varphi \Longrightarrow \Delta} \qquad (\neg D)\ \frac{\Gamma, \varphi \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta, \neg\varphi}$$

$$(\bot E)\ \frac{}{\Gamma, \bot \Longrightarrow \Delta} \qquad (\top D)\ \frac{}{\Gamma \Longrightarrow \Delta, \top}$$

Intuitivamente, um sequente $\gamma_1, \ldots, \gamma_n \Longrightarrow \delta_1, \ldots, \delta_k$ denota que

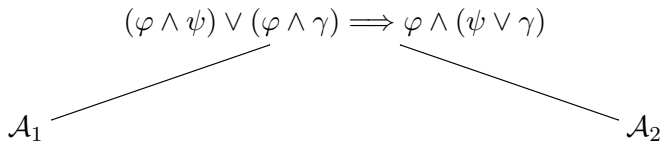
$$\bigwedge_{i=1}^{n} \gamma_i \vdash \bigvee_{i=1}^{k} \delta_i.$$

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), ($\bot$E) ou ($\top$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.

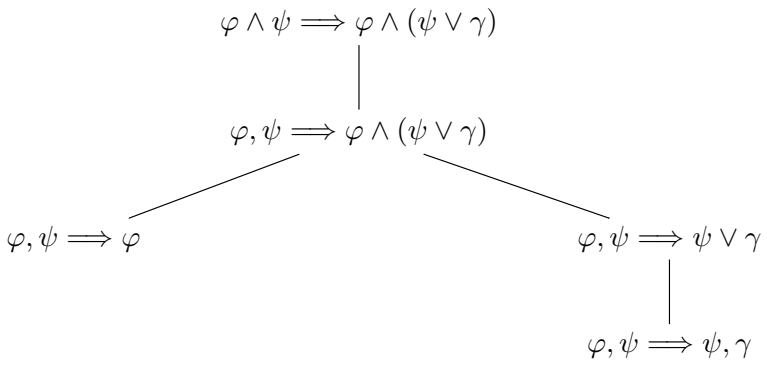
Exemplos 4.3.3.
(1) Provaremos o sequente $\varphi \wedge (\psi \vee \gamma) \Longrightarrow (\varphi \wedge \psi) \vee (\varphi \wedge \gamma)$ em SQ. Considere a seguinte derivação em SQ:

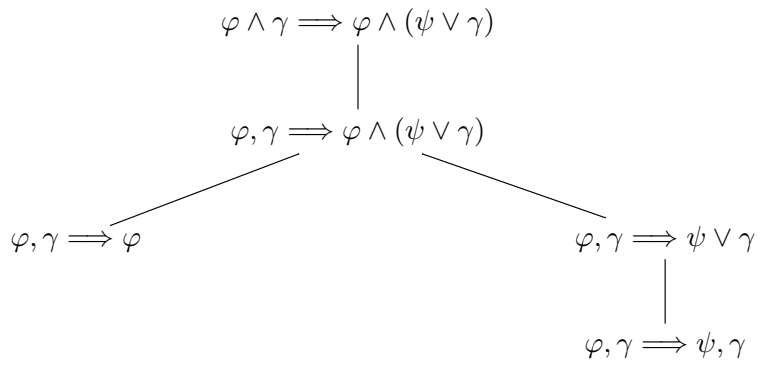
$$\dfrac{\dfrac{\dfrac{\varphi,\psi \Longrightarrow \varphi,\varphi\wedge\gamma \quad \varphi,\psi \Longrightarrow \psi,\varphi\wedge\gamma}{\varphi,\psi \Longrightarrow \varphi\wedge\psi,\varphi\wedge\gamma} \quad \dfrac{\varphi,\gamma \Longrightarrow \varphi,\varphi\wedge\psi \quad \varphi,\gamma \Longrightarrow \gamma,\varphi\wedge\psi}{\varphi,\gamma \Longrightarrow \varphi\wedge\psi,\varphi\wedge\gamma}}{\varphi,\psi\vee\gamma \Longrightarrow \varphi\wedge\psi,\varphi\wedge\gamma}}{\varphi\wedge(\psi\vee\gamma) \Longrightarrow \varphi\wedge\psi,\varphi\wedge\gamma}}{\varphi\wedge(\psi\vee\gamma) \Longrightarrow (\varphi\wedge\psi)\vee(\varphi\wedge\gamma)}$$

(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:



onde $\mathcal{A}_1$ e $\mathcal{A}_2$ são as seguintes árvores diádicas, respectivamente:





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:

Metateorema 4.3.4 (Hauptsatz)**.** *O sistema SQ^- obtido de SQ eliminando a regra (Corte) é equivalente com o sistema SQ.*

Observe que o cálculo de sequentes sem cortes SQ^-, se executado em sentido inverso como no Exemplo 4.3.3, resulta ser um método de prova analítico, bastante semelhante ao método de tablôs.

4.4 Exercícios

1. Verifique, usando o método de prova por tablôs, se as seguintes fórmulas são teoremas de PC.

   (a) $[(\varphi \to \psi) \to \psi] \to \psi$

   (b) $(\varphi \to \psi) \vee (\psi \to \varphi)$

   (c)

   (d)

   (e)

(f)

(g)

(h)

2. Usando o método de prova por tablôs, determine se o conjunto de fórmulas é consistente.

   (a) $\{\varphi \to \psi, \neg\psi \to \neg\varphi\}$

   (b) $\{(\varphi \wedge \psi) \to \gamma,\ (\varphi \to \gamma) \vee (\psi \to \gamma)\}$

   (c) $\{\varphi \leftrightarrow \psi,\ (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)\}$

   (d) $\{\varphi \to \gamma,\ \psi \to \gamma,\ (\varphi \vee \psi) \to \gamma\}$

   (e) $\{\alpha_0 \leftrightarrow \beta_0,\ \alpha_1 \leftrightarrow \beta_1,\ (\alpha_0 \to \alpha_1) \leftrightarrow (\beta_0 \to \beta_1)\}$

3. Escolha uma fórmula arbitrária e calcule sua FND usando tablôs.

4. Provar o seguinte utilizando tablôs, dedução natural e sequentes:

   (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) $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$.

Capítulo 5

Lógica de Predicados

Na teoria de Silogismos podemos distinguir dois tipos de proposições:

- Proposições gerais:
  - *todo homem é mortal*
  - *nenhum vegetal é um animal*
  - *alguns triângulos são isósceles*
- Proposições singulares:
  - *Sócrates é homem*
  - *meu carro é azul*
  - *5 é um número ímpar*

Observe que as proposições do primeiro tipo estabelecem certas relações entre classes, enquanto que as do segundo tipo afirmam (ou negam) a pertinência de um termo singular (ou indivíduo) a uma determinada classe. Esta distinção entre classes de indivíduos e indivíduos é a base das linguagens chamadas *de primeira ordem*, ou *linguagem de predicados*. Elas, munidas de uma lógica apropriada (*lógica de primeira ordem* ou *lógica de predicados*) constituem uma generalização natural da linguagem e da lógica silogística introduzida por Aristóteles. A concepção moderna da lógica de predicados deve-se a Frege. Veremos que as proposições gerais são obtidas das proposições singulares através do que chamaremos de *quantificação*, isto é, o uso de *quantificadores*. Os quantificadores que utilizaremos são os usuais: *para todo*, representado por $\forall$, e *existe*, representado por $\exists$.

Dado que a semântica das linguagens de primeira ordem está baseada na teoria de conjuntos (e, em particular, na teoria de relações e funções), na próxima seção faremos uma revisão sucinta desses tópicos.

5.1 Conjuntos, relações e funções

A teoria de conjuntos, introduzida por Georg Cantor em 1870, foi iniciada a partir das suas pesquisas sobre a teoria das séries infinitas em Análise; a partir destes trabalhos, Cantor foi levado a considerar conjuntos infinitos ou classes de caráter arbitrário. A idéia intuitiva é que um conjunto é uma coleção de objetos. Segundo Cantor, "um conjunto é uma coleção, considerada como um todo, de objetos distintos e definidos da nossa intuição ou pensamento. Os objetos são chamados de elementos do conjunto". A teoria axiomática de conjuntos é a base da matemática contemporânea: todos os conceitos matemáticos podem ser descritos em termos da linguagem dos conjuntos. A seguir daremos uma breve e sucinta revisão dos conceitos básicos mínimos da teoria axiomática de conjuntos, porém não enunciaremos, por enquanto, os axiomas de maneira formal. O motivo desta decisão é que, para expressar formalmente os axiomas da teoria de conjuntos, precisaríamos de uma linguagem de primeira ordem, que vai ser introduzida neste texto somente na próxima seção. Na Seção 5.4 daremos como exemplo alguns axiomas da Teoria Axiomática de Conjuntos conhecida como *sistema* ZF, denominada desta maneira em homenagem aos matemáticos Ernst Zermelo e Abraham Fränkel, que conseguiram formular de maneira correta (e livre de paradoxos) as idéias originais de Cantor. O sistema ZF é o principal sistema de teoria de conjuntos, porém existem outras formulações, tais como a teoria de conjuntos de John von Neumann, Paul Bernays e Kurt Gödel, conhecida como *sistema* NBG (denominado desta maneira em homenagem aos seus criadores).

Partimos então de uma noção intuitiva e primitiva, a de *conjunto*, e consideramos uma outra relação primitiva e intuitiva, a de *pertinência* de um conjunto a um outro conjunto. Se um conjunto, digamos A, pertence a outro conjunto, digamos B, diremos que A é um *elemento* de B. A sentença "A pertence a B" será denotada por $A \in B$. Se A não pertence a B escreveremos $A \notin B$.

O célebre paradoxo do lógico e filósofo Bertrand Russell mostra que, se for possível criar o conjunto $\mathcal{R}$ de todos os conjuntos A tais que $A \notin A$ então $\mathcal{R} \in \mathcal{R}$ sse $\mathcal{R} \notin \mathcal{R}$, uma contradição. Portanto devemos tomar cuidado com a propriedade de conjuntos P que utilizamos para criar o conjunto

$\{x \ : \ x$ satisfaz a propriedade $P\}$. Voltaremos depois sobre este ponto.

É importante observar que na Teoria de Conjuntos os únicos objetos que existem são os conjuntos. Desta maneira, não existe distinção entre os objetos que são *elementos* e os objetos que são *conjuntos* (coleção de elementos). Isto significa que um elemento A de um conjunto B é, por sua vez, um conjunto que possui elementos que, por sua vez, são conjuntos de elementos. Há uma única categoria de objetos na Teoria de Conjuntos: a categoria dos conjuntos. [1]

O princípio fundamental da Teoria de Conjuntos é o de *extensionalidade*: os conjuntos são dados exatamente pelos seus elementos. Em outras palavras, se A e B têm os mesmos elementos, então eles são iguais. Isto é: se

$$x \in A \text{ se e somente se } x \in B$$

para todo conjunto x, então podemos afirmar que $A = B$. Obviamente, se $A = B$ então A e B têm os mesmos elementos.

Observe que uma coleção de conjuntos que imediatamente podemos conceber é a coleção que não possui elementos. Uma tal coleção é um conjunto dito *vazio*. Pelo princípio de extensionalidade, o conjunto vazio é único (confira esta afirmação!). O conjunto vazio é denotado por $\emptyset$.

A partir da relação de pertinência, podemos definir uma outra relação entre conjuntos: a noção de *inclusão*. Dizemos que um conjunto A *está contido* num conjunto B, ou que A *é um subconjunto de* B, denotado por $A \subseteq B$, se vale o seguinte:

$x \in A$ implica que $x \in B$ para todo conjunto x.

isto é: $A \subseteq B$ sse todo elemento de A é um elemento de B. Observe que as relações de pertinência e de inclusão são diferentes. Por exemplo, podemos ter conjuntos A e B tais que $A \in B$ mas $A \not\subseteq B$, ou tais que $A \subseteq B$ mas $A \notin B$. Obviamente também podemos ter conjuntos A e B tais que $A \notin B$, $B \notin A$, $A \not\subseteq B$, $B \not\subseteq A$.

[1] No sistema NBG é feita uma distinção entre *conjuntos* e *classes*. As últimas representam coleções que podem ser muito grandes, e que portanto não podem ser completadas em nenhuma etapa. Os conjuntos são casos particulares de classes 'pequenas' que podem ser completadas em alguma etapa (mesmo que após de infinitos passos de construção). As classes que não são conjuntos são chamadas de *classes próprias*. Um exemplo elucidatório de classe própria é a coleção $\mathcal{V}$ de todos os conjuntos. Se $\mathcal{V}$ fosse um conjunto, então teriamos que $\mathcal{V} \in \mathcal{V}$, o que nos levaria a um paradoxo. Outra classe própria importante é a classe $\mathcal{R} = \{A \ : \ A$ é um conjunto e $A \notin A\}$. Se $\mathcal{R}$ fosse um conjunto então derivariamos o paradoxo de Russell. Assim, NBG tem duas categorias de objetos distintos: a categoria das classes próprias e a categoria dos conjuntos.

A lista inicial dos axiomas de ZF que mencionaremos aqui (expressados, como foi comentado anteriormente, numa linguagem informal), é a seguinte:

Axioma da Extensionalidade: Dois conjuntos são iguais sse têm os mesmos elementos.

Axioma de Separação: Se A é um conjunto e $\varphi(x)$ é uma propriedade de conjuntos, então existe o conjunto $\{x \ : \ x \in A \ \wedge \ \varphi(x)\}$ formado pelos elementos de A que satisfazem a propriedade φ.

Axioma do Par não-ordenado: Se A e B são conjuntos, então existe o conjunto $\{A, B\}$ que contém como elementos *exatamente* os conjuntos A e B.

Axioma da União: Se A é um conjunto então existe o conjunto $\bigcup A$ formado pelos elementos dos elementos de A.

Axioma do Conjunto das Partes: Se A é um conjunto então existe o conjunto $\wp(A)$ formado pelos conjuntos x tais que $x \subseteq A$.

A coleção $\{x \ : \ x \in A \ \wedge \ \varphi(x)\}$ pode ser denotada também por

$$\{x \in A \ : \ \varphi(x)\}.$$

A Teoria de Conjuntos nos permite *construir*, a partir dos axiomas, conjuntos a partir de outros conjuntos já construídos. Por exemplo:

- Existe o conjunto $\emptyset$ que não contém elementos: de fato, dado qualquer conjunto A, temos que $\emptyset = \{x \in A \ : \ x \neq x\}$, obtido portanto pelo axioma de separação.
- Se A é um conjunto então $\{A\}$ é um conjunto que contém um único elemento, o conjunto A (este tipo de conjuntos são ditos *unitários*). De fato, $\{A\}$ é obtido pelo axioma do par não-ordenado tomando $A = B$.
- Se A e B são conjuntos então existe o conjunto $A \cup B$ formado pelos conjuntos x tais que $x \in A$ ou $x \in B$ (união de conjuntos). De fato, $A \cup B = \bigcup\{A, B\}$ obtido pelo axioma do par e pelo axioma da união.
- Se A e B são conjuntos então existe o conjunto $A \cap B$ formado pelos conjuntos x tais que $x \in A$ e $x \in B$ (interseção de conjuntos). De fato $A \cap B = \{x \in A \ : \ x \in B\}$, obtido portanto pelo axioma de separação.

- Se A e B são conjuntos então existe o conjunto $A - B$ formado pelos conjuntos x tais que $x \in A$ e $x \notin B$ (diferença de conjuntos, ou complemento de B relativo a A). De fato $A - B = \{x \in A \ : \ x \notin B\}$, obtido portanto pelo axioma de separação.

Se uma coleção A consiste exatamente dos elementos $x_1, \ldots, x_n$ então podemos escrever $A = \{x_1, \ldots, x_n\}$.

O axioma de separação evita (ao menos em princípio) o paradoxo de Russell: nada garante que possamos criar a coleção $\mathcal{R} = \{x \ : \ x \notin x\}$ pois, para isso, deveriamos estar *separando* sobre algum conjunto. O axioma de separação permite, sim, a criação do conjunto $\mathcal{R}' = \{x \ : \ x \in A \ \wedge \ x \notin x\}$ que vai ser, de fato, idêntico com o conjunto A, uma vez que consideremos todos os axiomas de ZF, como veremos daqui a pouco.

A partir da notação que acabamos de introduzir, podemos mencionar outros dois importantes axiomas do sistema ZF:

Axioma da Regularidade: Se A é um conjunto não-vazio, então A contém um elemento B que é disjunto de A, isto é, tal que $A \cap B = \emptyset$.

Axioma do Infinito: Existe um conjunto A tal que $\emptyset \in A$ e, se $B \in A$, então $B \cup \{B\} \in A$.

O axioma da regularidade proibe situações indesejáveis como as seguintes:

- Que exista um conjunto A tal que $A \in A$;
- Que existam conjuntos A e B tais que $A \in B$ e $B \in A$;
- Em geral, que existam conjuntos $A_1, \ldots, A_n$ tais que

$$A_1 \in A_2 \in A_3 \in \cdots \in A_{n-1} \in A_n \in A_1 \text{ (ciclos).}$$

- Que exista uma família infinita de conjuntos $A_1, A_2, \ldots, A_n, \ldots$ tal que

$$\cdots \in A_{n+1} \in A_n \in A_{n-1} \in \cdots \in A_2 \in A_1 \text{ (descensos infinitos).}$$

Pela primeira das observações acima vemos que o conjunto de Russell admitido pelo axioma de separação, digamos $\mathcal{R}' = \{x \ : \ x \in A \ \wedge \ x \notin x\}$, é, de fato, o conjunto A (confira esta afirmação!).

Existem outros conjuntos que são proibidos de construir pelo uso do axioma da regularidade: por exemplo o conjunto

$$A = \{\{\{\cdots\}\}\}$$

tem um único elemento, o conjunto A, violando portanto o axioma da regularidade. Note que o único elemento de A é também o conjunto A, que por sua vez tem como único elemento o conjunto A, e assim *ad infinitum*. Em geral, o axioma da regularidade proibe a existência de um conjunto A tal que $A = \{A\}$. Porém, este axioma sozinho não impede a existência de um conjunto A tal que $A = \{\{B\}, A\}$ (se $B \neq A$). Se bem que $A \in A$ e $A \cap A \neq \emptyset$, temos por outro lado que $\{B\} \in A$ tal que $\{B\} \cap A = \emptyset$, pois o único elemento de $\{B\}$ é B, e $B \notin A$ (note que, dado que $A = \{\{B\}, A\}$, então $B \in A$ sse $B = \{B\}$ ou $B = A$; mas nenhuma destas possibilidades é verdadeira, portanto temos que $B \notin A$). Assim sendo, o conjunto A acima não está violando o axioma da regularidade. Porém, supor que existe um conjunto A tal que $A \in A$ *junto com* o axioma do par não-ordenado e o axioma da extensionalidade leva de fato a uma contradição. Com efeito, se $A \in A$ para algum conjunto A então o conjunto $\{A\}$ (que existe pelo axioma do par não-ordenado, como vimos anteriormente) viola o axioma da regularidade. Assim sendo, não existem conjuntos A tais que $A = \{\{B\}, A\}$, se considerarmos *todos* os axiomas da Teoria de Conjuntos. [2]

Por outro lado, o axioma do infinito estabelece, como indica seu nome, a existência de um conjunto *infinito*, isto é, com uma quantidade infinita de elementos. Concretamente, estabelece a existência de (uma representação de) o conjunto $\mathbb{N}$ dos números naturais. Com efeito, basta representar o número 0 como o conjunto $\emptyset$, e considerar a função sucessor $S(x) = x + 1$ como sendo $S(B) = B \cup \{B\}$. Assim, o número n é representando pelo conjunto $\mathbf{n} := \overbrace{S(\ldots S(}^{n \text{ vezes}}\emptyset)\ldots)$ (voltaremos sobre este assunto no Exemplo 5.2.4). Logo, $\mathbf{0} = \emptyset$ e $\mathbf{n+1} = \{\mathbf{0}, \mathbf{1}, \ldots, \mathbf{n}\}$. O axioma de infinito (junto com os outros axiomas) permite criar o conjunto $\omega = \{\mathbf{n} \ : \ n \in \mathbb{N}\}$, que representa o conjunto $\mathbb{N}$ dos números naturais. Sem este axioma, não é possível garantir a existência de conjuntos infinitos dentro do próprio sistema.

O sistema ZF consiste dos axiomas acima mencionados junto com um axioma, chamado de *Axioma da Substituição*, extremamente poderoso, e que tem, inclusive, como caso particular o axioma da separação. Dado que a sua

[2] Devemos destacar aqui que existem outras teorias alternativas de conjuntos que não pressupõem o axioma da regularidade, permitindo definir conjuntos *estranhos* como os acima mencionados, sem obter porém nenhuma contradição.

formulação é um pouco complexa, preferimos não escrevé-lo explicitamente. Finalmente, existe um outro axioma da Teoria de Conjuntos, chamado de *Axioma da Escolha* (chamado na literatura de (AC), por *Axiom of Choice*), que é bastante controverso. A controversia surge pelo caráter fortemente *não-construtivo* de (AC): ele estabelece, informalmente, o seguinte:

- Se A é um conjunto não-vazio tal que todos seus elementos são conjuntos não-vazios, então é possível escolher um elemento de cada elemento de A, formando um novo conjunto (axioma da escolha).

A não-construtividade intrínseca do (AC) está no fato de que não existe um critério (um algoritmo ou processo sitemático) dado *a priori* que nos permita escolher um elemento de uma coleção infinita. O seguinte exemplo intuitivo, devido a Russell, pode ajudar a entender o problema inerente de (AC):

Exemplo 5.1.1. Considere uma coleção A de pares de sapatos idênticos. Nesse caso, é possível *escolher* sem problemas um único elemento (sapato) de cada par de sapatos de A, por exemplo através do algoritmo "pegue apenas os sapatos que correspondam ao pé direito". Por outro lado, se agora consideramos uma coleção A' de pares de meias idênticas (não existindo nenhuma diferença entre a meia do pé direito e a do pé esquerdo), não existe um método imaginável que permita *escolher* apenas uma meia de cada par de A'. Porém, o Axioma da Escolha garante que é possível fazer isto! □

O sistema ZF pode ser estendido pelo acrêscimo de (AC), obtendo o sistema conhecido como ZFC. O (AC) tem suscitado fortes críticas por parte dos matemáticos, lógicos e filósofos que defendem a chamada *escola construtivista*, que é caracterizada, entre outras coisas, pelo fato de que para demonstrar a existência de um objeto satisfazendo certas características deve ser dada uma *construção* (realizada num número finito de passos) do objeto. Porém, o axioma da escolha é fundamental para o desenvolvimento da matemática moderna, como tem sido demonstrado por alguns lógicos e matemáticos: é possível provar que (AC) equivale, no contexto de ZF, a importantíssimos princípios fundamentais da matemática, sem os quais uma imensa parte da matemática contemporânea se perderia.

Uma vez que apresentamos ZF, vamos analisar alguns conceitos que serão relevantes para os nossos objetivos.

O axioma de separação permite expressar (e construir) os conjuntos da maneira usual em matemáticas, caracterizando uma coleção de elementos

pela propriedade que eles possuem em comum, no lugar de listar todos os elementos da coleção. Assim, podemos escrever indistintamente

$$\{3, 4, 5, 6\} \quad \text{ou} \quad \{x \in \mathbb{N} \ : \ 2 < x \leq 6\}.$$

Observemos que os conjuntos $\{a, b\}$ e $\{b, a\}$ são idênticos. Desta maneira, não podemos identificar uma ordem entre os elementos de um conjunto (e é por isso que o axioma chama-se de *axioma do par não-ordenado*). Se quisermos definir um conjunto de dois elementos estabelecendo uma ordem entre eles (o que, como veremos a seguir, é fundamental para o desenvolvimento da lógica de predicados) devemos definir o conceito de *par ordenado*.

Definição 5.1.2. Dados dois conjuntos A e B, o par ordenado (A, B) é o conjunto $\{\{A\}, \{A, B\}\}$. O conjunto A é a *primeira componente* do par, e B é a *segunda componente*. □

A propriedade fundamental dos pares ordenados é a seguinte: $(A, B) = (C, D)$ se e somente se $A = C$ e $B = D$. Observe que $(A, B) = (B, A)$ se e somente se $A = B$. Com este conceito podemos diferenciar entre os dois elementos de um conjunto, colocando eles em seqüência. Dados dois conjuntos A e B, podemos criar a coleção de todos os pares ordenados cuja primeira componente é um elemento de A e cuja segunda componente é um elemento de B. Esta coleção é o *produto cartesiano* de A e B, dado por

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

Uma vez que definimos o produto cartesiano, podemos estar interessados em escolher subconjuntos interessantes de $A \times A$. Um conjunto $R \subseteq A \times A$ é dita uma *relação em A*. Em geral, um subconjunto $R \subseteq A \times B$ é dita uma *relação*.

Por exemplo, dado o conjunto $\mathbb{N} = \{0, 1, 2, \ldots\}$ dos números naturais, a relação

$$R = \{(n, m) \in \mathbb{N} \times \mathbb{N} \ : \ n \leq m\}$$

é a relação de ordem entre números. Outro exemplo interessante de relação entre números naturais é a relação

$$S = \{(n, m) \in \mathbb{N} \times \mathbb{N} \ : \ n \ \text{ divide a } \ m\}.$$

Dada uma relação R, se $(a, b) \in R$ escreveremos aRb, como é usual em matemáticas.

Dizemos que uma relação R é:

- *reflexiva* se aRa para todo $a \in A$;
- *simétrica* se aRb implica que bRa;
- *asimétrica* se aRb implica que não é o caso que bRa;
- *antisimétrica* se aRb e bRa implica $a = b$;
- *transitiva* se aRb e bRc implica aRc;
- *de equivalência* se é reflexiva, simétrica e transitiva;
- *uma preordem* se é reflexiva e transitiva;
- *uma ordem parcial* se é reflexiva, antisimétrica e transitiva;
- *uma ordem total* se é uma ordem parcial tal que aRb ou bRa para todo $a, b \in A$.

O conceito de par ordenado pode ser generalizado, definindo *ternas ordenadas* (a, b, c) e, em geral, *n-uplas ordenadas* $(a_1, a_2, \ldots, a_n)$ em que a_1 é a primeira componente, a_2 é a segunda componente, ..., e a_n é a n-ésima componente. Podemos definir o produto cartesiano

$$A_1 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) \ : \ a_1 \in A_1 \ \text{ e } \ \cdots \ \text{ e } \ a_n \in A_n\}.$$

Uma *relação n-ária* é qualquer subconjunto R de $A_1 \times \cdots \times A_n$. Um subconjunto R de $\overbrace{A \times \cdots \times A}^{n \text{ vezes}}$ é uma *relação n-ária em A*. Por simplicidade, o conjunto $\overbrace{A \times \cdots \times A}^{n \text{ vezes}}$ será denotado por A^n. Veremos na próxima seção que as relações n-árias servem para interpretar propriedades e predicados em geral.

Finalmente, definimos o conceito de função, como caso particular de relação:

Definição 5.1.3. Uma *função f de A em B*, representada por $f : A \to B$, é uma relação $f \subseteq A \times B$ tal que, para todo a em A, existe um único $b \in B$ tal que $(a, b) \in f$. O elemento de B atribuído pela função f ao elemento a de A será denotado por $f(a)$. □

Por exemplo, a relação

$$f = \{(n, n^2) \ : \ n \in \mathbb{N}\}$$

é uma função $f : \mathbb{N} \to \mathbb{N}$ tal que $f(n) = n^2$ para todo $n \in \mathbb{N}$.

5.2 Linguagens de primeira ordem

Vamos considerar agora linguagens formais mais sofisticadas do que a linguagem da lógica proposicional. Estas linguagens são as *linguagens de primeira ordem* ou *linguagens de predicados*. A característica principal destas linguagens é partir da distinção entre *indivíduos* e *propriedades de indivíduos*. Trata-se de representar formalmente, e com certo grau de generalidade, as sentenças da linguagem natural da forma

sujeito–predicado.

Por exemplo,

Pedro é paulistano

é uma típica frase do tipo acima mencionado, em que o sujeito (indivíduo) é "Pedro" e o predicado (propiedade de indivíduos) é "(ser)paulistano". Um exemplo um pouco mais complexo é

Pedro e Maria são paulistanos.

Aqui o sujeito é composto de dois indivíduos: Pedro e Maria ou, mais formalmente, o par ordenado (Pedro,Maria).

A idéia básica da linguagem formal de predicados é a de expressar de maneira geral este tipo de sentenças (e outras mais complexas, como veremos a seguir). Partimos da premissa de que há certas propriedades de indivíduos básicas, dadas *a priori*, e que determinam a base da linguagem formal, a partir da qual poderão ser construídas as sentenças mais complexas. Estas propriedades básicas são chamadas de *predicados*. Assim, nos dois exemplos acima, a propriedade básica é a de ser paulistano. Se quisermos escrever então as duas sentenças acima numa linguagem de primeira ordem, precisariamos de um símbolo para o predicado "(ser)paulistano", digamos `Paulistano`, e de símbolos para os indivíduos "Pedro"e "Maria", digamos `pe` e `ma`. Desta maneira, a primeira sentença poderia ser escrita como `Paulistano(pe)`. A partir de este tipo de expressões formais (chamadas de *fórmulas atômicas*) podemos construír, pelo uso dos mesmos conectivos lógicos das linguagens proposicionais, formar sentenças formais (chamadas de *fórmulas complexas*). No caso do segundo exemplo acima, a partir das fórmulas atômicas `Paulistano(pe)` e `Paulistano(ma)` podemos formalizar, pelo uso do conectivo da conjunçao (simbolizado por $\wedge$) a fórmula complexa

$$\texttt{Paulistano(pe)} \wedge \texttt{Paulistano(ma)}$$

que representa, numa linguagem formal de primeira ordem apropriada, a sentença em português *Pedro e Maria são paulistanos*. Podemos dar um exemplo de formalização um pouco mais sofisticado: a sentença

Se Pedro é corintiano e João é sãopaulino, então Pedro não gosta de João

pode ser formalizada utilizando três símbolos de predicado, digamos `Cori`, `Saop` e `Nao_gosta`. Estes predicados simbolizam as propriedades de ser corintiano, ser sãopaulino e a relação *não gostar de*. Observe que esta última relação é *binária*, isto é, relaciona dois indivíduos. Assim, `Nao_gosta(x,y)` é a expressão formal para dizar que o indivíduo x não gosta do indivíduo y. Finalmente, utilizamos dois símbolos de constante `pe` e `jo` para representar os indivíduos "Pedro" e "João". A sentença acima pode ser então representada pela fórmula complexa

$$(\texttt{Cori(pe)} \wedge \texttt{Saop(jo)}) \rightarrow \texttt{Nao\_gosta(pe,jo)}$$

em que $\rightarrow$ é o símbolo para a implicação material. Poderiamos refinar ainda mais a representação anterior, considerando como propriedade básica a relação *gostar de*, representada pelo predicado predicado binário `Gosta`. Desta maneira, a sentença acima pode ser representada pela fórmula complexa

$$(\texttt{Cori(pe)} \wedge \texttt{Saop(jo)}) \rightarrow \neg\texttt{Gosta(pe,jo)} \qquad (1)$$

em que $\neg$ é o conectivo da negação.

Os predicados são utilizados para representar *relações* entre objetos (indivíduos). Uma relação é unária, binária ou ternária se relaciona um, dois ou três indivíduos. Assim, "(ser)italiano", "(ser)azul", "(ser)triângulo" são exemplos de relações unárias (as que chamamos de *propriedades*); "amigo de", "gosta de", "pai de" são exemplos de relações binárias. Na sentença

Jundiaí fica entre São Paulo e Campinas

encontramos um exemplo de relação ternária, "entre": nesse caso, podemos definir um predicado ternário `Entre` tal que $\texttt{Entre}(a, b, c)$ é o caso se e somente se a fica entre b e c. Outros exemplo de relações ternárias são dados nas sentenças

Helena apresentou João a Maria

João ganhou um livro da Maria

que podem ser formalizadas pelas fórmulas

`Apresentou(helena,joao,maria)` e `Ganhou(joao,livro,maria)`,

respectivamente.

Obviamente existem relações de mais de três argumentos, digamos relações vinculando n indivíduos. Como vimos na Seção 5.1, estas são relações n-árias. Dada uma relação n-ária `P`, a fórmula $\texttt{P}(i_1, \ldots, i_n)$ indica que os indivíduos $i_1, \ldots, i_n$ estão relacionados através de `P`.

Exemplos de relações 4-árias aparecem nas seguintes sentenças:

João comprou um iPod da Maria por 300 dólares

João, Pedro, Luiz e Arnaldo jogaram truco

que podem ser representadas pelas fórmulas

`Comprou(j,iP,m,300)` e `Jogaram_truco(joao,pedro,luiz,arnaldo)`,

respectivamente.

É fundamental observar que a *ordem* dos argumentos numa relação n-ária (com $n \geq 2$) é relevante: Dada a sentença (verdadeira) *Platão foi mestre de Aristóteles*, podemos formalizá-la pela fórmula `Mestre(p,a)`. Nesta linguagem de predicados, a fórmula `Mestre(a,p)` formaliza a sentença (falsa) *Aristóteles foi mestre de Platão*. Analogamente, $5 \leq 3$ tem significado diferente de $3 \leq 5$. Isso significa que a ordem dos argumentos é relevante para o significado da sentença que está sendo representada. Assim, na expressão $\texttt{P}(i_1, \ldots, i_n)$ estamos assumindo que os indivíduos $i_1, \ldots, i_n$ estão dados em seqüência, isto é, os argumentos do predicado `P` são dados pela n-upla $(i_1, \ldots, i_n)$ e não pelo conjunto $\{i_1, \ldots, i_n\}$ (veja a Seção 5.1). Outros exemplos em que a ordem dos argumentos é relevante são os seguintes:

João é o pai de Carlos	`pai(j,c)`
João ama Maria	`ama(j,m)`

Voltemos agora ao exemplo (1) dos corintianos e sãopaulinos. Suponhamos que queremos expressar uma situação mais forte do que (1): não apenas Pedro não gosta de João pelo fato dele ser corintiano e o João ser sãopaulino, senão que *todo corintiano* e *todo sãopaulino* estão em enemizade. A sentença

Se Pedro é corintiano e João é sãopaulino, então eles são inimigos

poderia ser representada formalmente pela fórmula complexa

$$(\texttt{Cori(pe)} \wedge \texttt{Saop(jo)}) \rightarrow \texttt{Inimigo(pe,jo)} \qquad (2)$$

procedendo de maneira análoga. Aqui estamos supondo, por simplicidade, que a relação de enemizade é simétrica. Assim, a situação mais forte (em que todos os corintianos e sãopaulinos são inimigos) é expressada, na linguagem natural, pela sentença

Os corintianos e os sãopaulinos são inimigos

ou, mais explicitamente,

> *Se uma pessoa, digamos x, é corintiana, e outra pessoa, digamos y, é sãopaulina, então x e y são inimigos.*

Noutras palavras,

> *Dado um indivíduo x e dado um indivíduo y, se x é corintiano e y é sãopaulino então x e y são inimigos*

ou, equivalentemente,

> *para todo indivíduo x e para todo indivíduo y, se x é corintiano e y é sãopaulino então x e y são inimigos.*

Neste ponto identificamos a estrutura lógica da sentença original, e podemos expressá-la numa linguagem de primeira ordem através da fórmula complexa

$$(\forall x)(\forall y)((\texttt{Cori}(x) \wedge \texttt{Saop}(y)) \rightarrow \texttt{Inimigo}(x, y)) \qquad (3).$$

O senso comum indica que a fórmula (2) deveria ser uma *conseqüência lógica* da fórmula (3). Este é o caso, com efeito, como veremos depois.

Na fórmula (3) utilizamos dois tipos novos de símbolos: *variáveis individuais* ou *variáveis de indivíduo* (x e y) e *quantificadores universais* ($\forall x$ e $\forall y$). O significado das expressões utilizando estes símbolos é o esperado: as variáveis representam indivíduos arbitrários, gerais, sobre os quais não temos nenhuma informação prévia. Assim, $\texttt{Paulistano}(x)$ denota a afirmação *o indivíduo x é paulistano*. Esta afirmação carece de valor de verdade se não instanciamos a variável x em algum indivíduo concreto.

As variáveis são necessárias para poder expressar convenientemente a quantificação. Uma fórmula φ em que aparece a variável x sem quantificar é

escrita como $\varphi(x)$, para pôr em evidência que φ representa uma propriedade sobre o indivíduo x. Uma fórmula $\varphi(x)$ poderia ser pensada como sendo um *predicado complexo*, para diferenciá-lo dos predicados básicos da linguagem. Da mesma maneira, a fórmula φ dada por

$$((\texttt{Cori}(x) \wedge \texttt{Saop}(y)) \rightarrow \texttt{Inimigo}(x, y))$$

poderia ser escrita como $\varphi(x, y)$, dado que as variáveis x e y aparecem sem quantificar. Novamente, uma fórmula $\varphi(x, y)$ poderia ser pensada como sendo um predicado complexo, neste caso de duas variáveis, representando portanto uma relação binária. Obviamente podemos definir predicados complexos de qualquer aridade.

Uma fórmula da forma $(\forall x)\varphi(x)$ representa a afirmação de que *todos* os indivíduos satisfazem o predicado (complexo) representado pela fórmula $\varphi(x)$. A fórmula (3), que é da forma $(\forall x)(\forall y)\varphi(x, y)$, representa a afirmação de que todos os pares de indivíduos (x, y) satisfazem o predicado (complexo) $\varphi(x, y)$.

Deveria ser observado que, desde esta perspectiva, as linguagens de primeira ordem tratam de frases da forma *sujeito–predicado* de uma maneira geral. Assim, $\varphi(x, y)$ seria (a formalização de) uma frase deste tipo, em que (x, y) é o sujeito e φ é o predicado. Na fórmula

$$\texttt{Paulistano(pe)} \wedge \texttt{Paulistano(ma)}$$

o sujeito é o par ordenado `(pe,ma)` e o predicado binário (complexo) é

$$\texttt{Paulistano}(x) \wedge \texttt{Paulistano}(y).$$

No exemplo clássico do famoso silogismo aristotélico que contém a premissa

Todo homem é mortal

podemos representar esta sentença pela fórmula complexa

$$(\forall x)(\texttt{Homem}(x) \rightarrow \texttt{Mortal}(x)).$$

A fórmula $(\texttt{Homem}(x) \rightarrow \texttt{Mortal}(x))$ é do tipo *sujeito–predicado*, em que x é o sujeito e o predicado é a propriedade "não ser homem ou ser mortal".

O outro tipo de quantificação que consideraremos na lógica de primeira ordem é a *quantificação existencial*. Esta quantificação aparece quando queremos afirmar que *algum* indivíduo (não necessariamente *todos*) satisfaz uma dada propriedade. Isto formaliza a expressão "algum", "alguns", "alguem",

"existe", "existem" etc. A notação utilizada é $(\exists x)\varphi(x)$, para uma dada fórmula $\varphi(x)$. Por exemplo, se queremos representar a sentença *existem corintianos*, ou seja, *existe um indivíduo x tal que x é corintiano*, podemos utilizar a fórmula

$$(\exists x)\texttt{Cori}(x).$$

Para expressar a sentença *alguém gosta de Maria* numa linguagem de primeira ordem poderiamos escrever a seguinte fórmula:

$$(\exists x)\texttt{Gosta}(x, \texttt{ma}).$$

representando a sentença em português acima.

Sentenças mais complexas podem ser representadas, combinando os conectivos e os quantificadores. Por exemplo

Se uma pessoa tem alguem que gosta dela, então essa pessoa é feliz

pode ser adequadamente representada pela fórmula

$$(\forall x)((\exists y)\texttt{Gosta}(y, x) \to \texttt{Feliz}(x)).$$

Observe que a fórmula acima é a quantificação universal da fórmula

$$\varphi(x) := ((\exists y)\texttt{Gosta}(y, x) \to \texttt{Feliz}(x)).$$

Esta última fórmula, por sua vez, é formada por uma implicação material em que o antecedente representa "alguém gosta do indivíduo x" enquanto que o consequente representa "o indivíduo x é feliz".

Nas combinações entre os quantificadores $\forall$ e $\exists$ devemos tomar o mesmo cuidado que tomamos com a ordem dos argumentos nos predicados: a ordem em que os quantificadores são colocados muda radicamente o sentido da frase a ser representada. Assim,

$$(\forall x)(\exists y)\texttt{Gosta}(x, y) \quad \text{e} \quad (\exists y)(\forall x)\texttt{Gosta}(x, y)$$

são fórmulas representando as sentenças em português *todo mundo gosta de alguém* e *alguém gosta de todo mundo*, respectivamente. Obviamente o sentido de estas duas sentenças é radicalmente diferente. Na Seção 5.3 provaremos rigorosamente que estas sentenças não são equivalentes, do ponto de vista da lógica (veja a observação após o Exemplo 5.3.8).

Uma vez que demos esta introdução informal às linguagens de primeira ordem, podemos defini-las formalmente.

Definição 5.2.1. Uma *assinatura de primeira ordem* é uma terna de conjuntos $\mathcal{S} = \langle \mathcal{P}, \mathcal{F}, \mathcal{C} \rangle$. Os elementos de $\mathcal{P}$, $\mathcal{F}$ e $\mathcal{C}$ são chamados de *símbolos de predicados*, *símbolos de funções* e *constantes*, respectivamente. Assumimos que cada símbolo de predicado e de constante tem associado um número natural $n \geq 1$, a *aridade* do símbolo. □

Assumiremos fixado um repertório inesgotável de símbolos (chamados de *variáveis individuais*) $\{x_1, x_2, \ldots\}$. Com uma assinatura (junto com as variáveis individuais, os conectivos e os quantificadores) podemos descrever os *indivíduos* e as *fórmulas* que conformarão a linguagem de primeira ordem gerada pela assinatura.

Definição 5.2.2. Dada uma assinatura de primeira ordem $\mathcal{S} = \langle \mathcal{P}, \mathcal{F}, \mathcal{C} \rangle$, definimos o conjunto dos *termos de* $\mathcal{S}$ como sendo o conjunto $\mathsf{Ter}(\mathcal{S})$ satisfazendo as seguintes cláusulas:

1. se x é uma variável individual então $x \in \mathsf{Ter}(\mathcal{S})$;
2. se c é uma constante de $\mathcal{C}$ então $c \in \mathsf{Ter}(\mathcal{S})$;
3. se f é um símbolo de função de aridade n de $\mathcal{F}$ e $t_1, \ldots, t_n$ pertencem a $\mathsf{Ter}(\mathcal{S})$ então $f(t_1, \ldots, t_n) \in \mathsf{Ter}(\mathcal{S})$;
4. os únicos elementos de $\mathsf{Ter}(\mathcal{S})$ são aqueles definidos pelas cláusulas anteriores. □

Os elementos de $\mathsf{Ter}(\mathcal{S})$ são chamados de *termos de* $\mathcal{S}$, e representam os indivíduos que podemos exibir concretamente na assinatura $\mathcal{S}$. As propriedades dos indivíduos (ou, em geral, as relações entre eles) são expressadas pelas *fórmulas*, cf. a definição seguinte.

Definição 5.2.3. Dada uma assinatura de primeira ordem $\mathcal{S} = \langle \mathcal{P}, \mathcal{F}, \mathcal{C} \rangle$, definimos o conjunto das *fórmulas de* $\mathcal{S}$ como sendo o conjunto $\mathsf{For}(\mathcal{S})$ satisfazendo as seguintes cláusulas:

1. se P é um símbolo de predicado de aridade n de $\mathcal{P}$ e $t_1, \ldots, t_n$ são termos então $P(t_1, \ldots, t_n) \in \mathsf{For}(\mathcal{S})$ (fórmulas atômicas);
2. se φ e ψ pertencem a $\mathsf{For}(\mathcal{S})$ então $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$ e $(\varphi \to \psi)$ pertencem a $\mathsf{For}(\mathcal{S})$;
3. se φ pertence a $\mathsf{For}(\mathcal{S})$ então $\neg\varphi$ pertence a $\mathsf{For}(\mathcal{S})$;

4. se φ pertence a $\mathsf{For}(\mathcal{S})$ e x é uma variável então $(\forall x)\varphi$ e $(\exists x)\varphi$ pertencem a $\mathsf{For}(\mathcal{S})$;

5. os únicos elementos de $\mathsf{For}(\mathcal{S})$ são aqueles definidos pelas cláusulas anteriores. □

Exemplo 5.2.4. Suponhamos que queremos definir uma linguagem de primeira ordem para falar da aritmética. Os símbolos que precisariamos introduzir na assinatura (que chamaremos de $\mathcal{S}_{ar}$) são os seguintes:

- uma relação binária $\preceq$ para representar a ordem $\leq$ entre os números naturais;
- uma relação binária $\approx$ para representar a igualdade entre os números naturais;
- uma constante $\mathsf{0}$ para representar o número 0;
- um símbolo de função unária s para representar a *função sucessor* dada por $S(n) = n + 1$;
- um símbolo de função binária $\oplus$ para representar a função de adição "+";
- um símbolo de função binária $\odot$ para representar a função de multiplicação "·".

Por simplicidade, e por analogia com a prática matemática, escreveremos $t_1 \oplus t_2$, $t_1 \odot t_2$, $t_1 \preceq t_2$ e $t_1 \approx t_2$ no lugar de $\oplus(t_1, t_2)$, $\odot(t_1, t_2)$, $\preceq (t_1, t_2)$ e $\approx (t_1, t_2)$, respectivamente, para termos t_1 e t_2 quaisquer.

Nesta assinatura, o número 3 é representado por exemplo pelo termo $s(s(s(\mathsf{0})))$. Em geral, o número n é representado pelo termo $\overbrace{s(\ldots s(}^{n \text{ vezes}}\mathsf{0})\ldots)$. Por outro lado, as sentenças da aritmética

$$0 \neq 1, \quad 2 + 2 = 4, \quad 1 \leq 3, \quad 2.3 \not\leq 5$$

são representadas pelas fórmulas

$$\neg(\mathsf{0} \approx s(\mathsf{0})), \quad s(s(\mathsf{0})) \oplus s(s(\mathsf{0})) \approx s(s(s(s(\mathsf{0})))), \quad s(\mathsf{0}) \preceq s(s(s(\mathsf{0}))),$$

$$\neg(s(s(\mathsf{0})) \odot s(s(s(\mathsf{0}))) \preceq s(s(s(s(s(s(\mathsf{0})))))),$$

respectivamente. Uma frase mais complexa, tal como "x é múltiplo de 3", é representada utilizando a assinatura $\mathcal{S}_{ar}$ pela fórmula

$$(\exists y)(x \approx s(s(s(\mathsf{0}))) \odot y).$$

Analogamente, a relação binária *divide a* entre números naturais é representada pela fórmula

$$\varphi(x, y) := (\exists z)(x \odot z \approx y)$$

tal que $\varphi(x, y)$ é o caso sse x divide a y. □

Exemplo 5.2.5. Suponhamos que queremos representar formalmente o seguinte argumento válido:

Todo homem é um animal.

Portanto, a cabeça de um homem é a cabeça de um animal.

Consideremos uma assinatura $\mathcal{S}$ contendo dois símbolos de predicados unários `Homem` e `Animal`, e um símbolo de predicado binário `Cabeça` tal que a fórmula $\mathtt{Cabeça}(x, y)$ representa a relação "x é cabeça de y". A premissa do argumento acima é representada pela fórmula

$$(\forall x)(\mathtt{Homem}(x) \to \mathtt{Animal}(x)).$$

Para representar a conclusão do argumento acima precisamos escrever algumas fórmulas intermediárias. A frase "x é a cabeça de um homem" é representada pela fórmula

$$\varphi(x) := (\exists y)(\mathtt{Cabeça}(x, y) \wedge \mathtt{Homem}(y)).$$

Analogamente, a frase "x é a cabeça de um animal" é representada pela fórmula

$$\psi(x) := (\exists y)(\mathtt{Cabeça}(x, y) \wedge \mathtt{Animal}(y)).$$

Portanto, a conclusão do argumento pode ser representada pela fórmula $(\forall x)(\varphi(x) \to \psi(x))$. A partir de aqui, a validade do argumento acima pode ser testada rigorosamente com os métodos semânticos que serão introduzidos na Seção 5.3. □

5.3 Semântica das linguagens de predicados

Nesta seção introduziremos a semântica para as linguagens de primeira ordem definidas na seção anterior. Esta noção de verdade foi introduzida por Tarski, e pode ser reduzida ao exemplo seguinte:

'a neve é branca' é verdadeira (numa interpretação) se e somente se a neve é branca (nessa interpretação)

Esta tautologia aparente deve ser entendida: quando tentamos dar um significado para a frase *a neve é branca*, devemos primeiro entender a sua sintaxe (gramática). Logo devemos interpretar (dar um significado) a cada um dos seus símbolos, e depois, a partir da sua gramática, obter mediante um procedimento recursivo o valor de verdade da frase a partir do valor de verdade dado às suas componentes.

No caso da sentença *a neve é branca*, a gramática é muito simples: trata-se de uma frase do tipo *sujeito–predicado*, logo ela é verdadeira numa dada interpretação se e somente se a interpretação do sujeito (que resulta ser um indivíduo concreto do universo da interpretação) satisfaz a interpretação do predicado (que, por sua vez, é uma propriedade concreta na estrutura da interpretação). Noutras palavras, o termo singular 'neve' deveria ser interpretado numa estrutura semântica como sendo um indivíduo ou sujeito ou objeto concreto $\mathbf{n}$ do universo de discurso U da estrutura. Por outro lado, o predicado 'branco' deveria ser interpretado semanticamente (e numa perspectiva extensional) como sendo uma propriedade dentro da estrutura, ou seja, um subconjunto B do universo de discurso U da estrutura: B seria o conjunto de todos os objetos de U que tem a propriedade (concreta) de serem brancos. Portanto a sentença *a neve é branca* é verdadeira na estrutura dada se e somente se $\mathbf{n} \in B$ é o caso na estrutura, isto é, o objeto 'neve' (da estrutura) satisfaz a propriedade 'ser branco' (da estrutura).

Fórmulas mais complexas (envolvendo o uso de conectivos e de quantificadores) serão interpretadas de maneira recursiva, seguindo as instruções específicas associadas a cada conectivo e cada quantificador.

Formalmente temos o seguinte:

Definição 5.3.1. Seja $\mathcal{S}$ uma assinatura de primeira ordem. Uma *estrutura para* $\mathcal{S}$ (ou uma *interpretação para* $\mathcal{S}$, ou uma $\mathcal{S}$*-estrutura*) é um par $\mathfrak{A} = \langle D, I \rangle$ em que D é um conjunto não-vazio (o *universo* da interpretação) e I é uma função de interpretação que atribui significado aos símbolos da assinatura $\mathcal{S}$ da maneira seguinte:

- se c é uma constante então $I(c)$ é um elemento de D;
- se f é um símbolo de função de aridade n então $I(f)$ é uma função $I(f) : D^n \to D$;
- se P é um símbolo de predicado de aridade n então $I(P)$ é uma relação n-ária $I(P) \subseteq D^n$.

O conjunto D é chamado de *domínio* ou *universo* da estrutura $\mathfrak{A}$. □

Observe que uma estrutura $\mathfrak{A}$ dá um significado concreto aos símbolos de $\mathcal{S}$. A partir daí é possível dar um significado concreto aos termos de $\mathcal{S}$ que não contêm variáveis (este tipo de termos são chamados de *termos fechados*).

Assim, se t é um termo fechado de $\mathcal{S}$ então definimos um elemento de D, denotado por $t^{\mathfrak{A}}$, de maneira recursiva:

- se t é uma constante então $t^{\mathfrak{A}} = I(t)$;
- se t é da forma $f(t_1, \ldots, t_n)$ então $t^{\mathfrak{A}} = I(f)(t_1^{\mathfrak{A}}, \ldots, t_n^{\mathfrak{A}})$.

Usando a possibilidade de "concretizar os termos" como sendo indivíduos de uma certa estrutura, definiremos a seguir como interpretar fórmulas numa dada estrutura. O objetivo é dar um valor de verdade (verdadeiro ou falso) a cada fórmula. Isto permitirá o desenvolvimento de uma lógica, chamada de *lógica de primeira ordem* ou *lógica de predicados*. Para isso devemos nos restringir a fórmulas nas quais toda variável aparece quantificada. Assim fórnulas tais como $(\forall x)(P(x, y)$, $(\exists x)(\forall y)P(x, y) \to P(x, c)$ ou $(\exists x)P(x) \vee P(x)$ não serão consideradas para fins semânticos. Observe que nestas fórmulas algumas variáveis não estão quantificadas. No primeiro exemplo, a variável y não está quantificada. No segundo e terceiro exemplo, a terceira ocorrência de x (de esquerda à direita) não está quantificada. Uma ocorrência não quantificada de uma variável numa fórmula chama-se de *ocorrência livre*. Caso contrário, é uma *ocorrência ligada*. Observe que uma variável pode ocorrer várias vezes numa fórmula, algumas ocorrências sendo livres e outras ligadas. Uma fórmula que não registra ocorrências livres de variáveis é chamada de *fórmula fechada* ou *sentença* (não confundir com as sentenças da linguagem natural).

Mais ainda, para poder interpretar os quantificadores numa dada estrutura, precisaremos estender a assinatura original acrescentando uma constante nova para cada elemento do domínio da interpretação. Isto justifica a definição seguinte:

Definição 5.3.2. Seja $\mathcal{S}$ uma assinatura de primeira ordem, e seja $\mathfrak{A} = \langle D, I\rangle$ uma estrutura para $\mathcal{S}$.
(ii) A *assinatura $\mathcal{S}$ estendida por $\mathfrak{A}$* é a assinatura $\mathcal{S}_{\mathfrak{A}}$ obtida de $\mathcal{S}$ acrescentando um novo símbolo de constante c_a para cada elemento a do domínio D de $\mathfrak{A}$.
(ii) A *estrutura derivada de $\mathfrak{A}$* é a $\mathcal{S}_{\mathfrak{A}}$-estrutura $\mathfrak{A}_D = \langle D, I_D\rangle$ tal que I_D interpreta os símbolos de $\mathcal{S}$ da mesma maneira que I, e $I_D(c_a) = a$ para cada $a \in D$. □

Isto significa que em $\mathcal{S}_{\mathfrak{A}}$ damos um "nome" (uma constante) c_a para cada elemento a de D. Por outro lado, na estrutura $\mathfrak{A}_D$ interpretamos o nome (constante) c_a de cada elemento a de D como sendo o próprio a. Isto parece ser um mero jogo formal, mas na realidade é uma técnica muito útil, que nos permitirá definir a semântica dos quantificadores, como veremos a seguir. [3]

Finalmente, dadas uma constante c e uma fórmula $\psi(x)$ em que a única variável que ocorre livre é possivelmente x, então $\psi(c)$ denota a sentença obtida de $\psi(x)$ substituindo cada ocorrência livre de x por c. Podemos agora definir a semântica das fórmulas:

Definição 5.3.3. Seja $\mathcal{S}$ uma assinatura de primeira ordem, e seja $\mathfrak{A} = \langle D, I\rangle$ uma estrutura para $\mathcal{S}$. Considere a assinatura $\mathcal{S}_{\mathfrak{A}}$ e a estrutura $\mathfrak{A}_D$ para $\mathcal{S}_{\mathfrak{A}}$ definidas na Definição 5.3.2. Dizemos que a estrutura $\mathfrak{A}_D$ *satisfaz uma sentença φ de $\mathcal{S}_{\mathfrak{A}}$*, denotado por $\mathfrak{A}_D \vDash \varphi$, se vale o seguinte:

1. $\varphi = P(t_1, \ldots, t_n)$ é uma sentença atômica, para um símbolo de predicado n-ário P e termos fechados $t_1, \ldots, t_n$. Então $\mathfrak{A}_D \vDash P(t_1, \ldots, t_n)$ se e somente se $(t_1^{\mathfrak{A}_D}, \ldots, t_n^{\mathfrak{A}_D}) \in I(P)$;

2. $\varphi = \neg\psi_1$. Então $\mathfrak{A}_D \vDash \neg\psi_1$ se e somente se $\mathfrak{A}_D \nvDash \psi_1$;

3. $\varphi = (\psi_1 \wedge \psi_2)$. Então $\mathfrak{A}_D \vDash (\psi_1 \wedge \psi_2)$ se e somente se $\mathfrak{A}_D \vDash \psi_1$ e $\mathfrak{A}_D \vDash \psi_2$;

4. $\varphi = (\psi_1 \vee \psi_2)$. Então $\mathfrak{A}_D \vDash (\psi_1 \vee \psi_2)$ se e somente se $\mathfrak{A}_D \vDash \psi_1$ ou $\mathfrak{A}_D \vDash \psi_2$;

5. $\varphi = (\psi_1 \to \psi_2)$. Então $\mathfrak{A}_D \vDash (\psi_1 \to \psi_2)$ se e somente se $\mathfrak{A}_D \nvDash \psi_1$ ou $\mathfrak{A}_D \vDash \psi_2$;

[3] A técnica de introduzir constantes novas para nomear cada um dos elementos do domínio de uma estrutura é conhecido como *método das constantes*, e é uma das técnicas fundamentais na área da lógica denominada Teoria de Modelos.

6. $\varphi = (\forall x)\psi$. Então $\mathfrak{A}_D \vDash (\forall x)\psi$ se e somente se $\mathfrak{A}_D \vDash \psi(c_a)$ para todo $a \in D$;

7. $\varphi = (\exists x)\psi$. Então $\mathfrak{A}_D \vDash (\exists x)\psi$ se e somente se $\mathfrak{A}_D \vDash \psi(c_a)$ para algum $a \in D$. □

Observe que toda sentença de $\mathcal{S}$ é, em particular, uma sentença de $\mathcal{S}_{\mathfrak{A}}$. Desta maneira, definimos o seguinte:

Definição 5.3.4. Seja $\mathcal{S}$ uma assinatura de primeira ordem, e seja $\mathfrak{A}$ uma estrutura para $\mathcal{S}$. Dizemos que $\mathfrak{A}$ *satisfaz uma sentença* φ *de* $\mathcal{S}$, denotado por $\mathfrak{A} \vDash \varphi$, se $\mathfrak{A}_D \vDash \varphi$. Uma sentença é *satisfatível* se existe alguma estrutura que a satisfaz. Finalmente, uma sentença φ é *universalmente válida* (ou simplesmente *válida*) se φ é satisfeita por toda estrutura $\mathfrak{A}$. □

Neste ponto é importante observar várias coisas:

1. No item 1 da Definição 5.3.3, estamos formalizando a noção de verdade de Tarski; uma frase do tipo *sujeito–predicado* é satisfeita por uma estrutura se e somente se a interpretação do sujeito nessa estrutura satisfaz a propriedade obtida pela interpretação do predicado nessa estrutura.

2. Os items 2 a 5 da Definição 5.3.3 refletem o caráter Booleano dos conectivos lógicos. Ou seja, a lógica de predicados *estende* a lógica proposicional clássica.

3. Nos items 6 e 7 reproduzimos a essência dos quantificadores. Uma sentença $(\forall x)\psi(x)$ é satisfeita por uma estrutura se e somente se *cada* elemento do domínio da estrutura satisfaz a propriedade denotada por $\psi(x)$. Por outro lado, a sentença $(\exists x)\psi(x)$ é satisfeita pela estrutura se e somente se *algum* elemento do domínio da estrutura satisfaz a propriedade denotada por $\psi(x)$. Dado que precisamos testar $\varphi(x)$ em *todos* os elementos de D (tanto para o quantificador $\forall$ quanto para o quantificador $\exists$), vemos a necessidade de introduzir um "nome" (uma constante) para cada elemento de D. Isso justifica a mudança da assinatura, de $\mathcal{S}$ para $\mathcal{S}_{\mathfrak{A}}$. E daí, obviamente precisamos estender $\mathfrak{A}$ para $\mathfrak{A}_D$. Vemos assim que, o que em princípio parecia ser um jogo formal, acabou se revelando uma técnica extremamente útil para dar conta da semântica dos quantificadores.

Uma vez que contamos com uma noção de *satisfação* de sentenças numa dada estrutura, podemos definir o conceito de *conseqüência lógica*, dando origem à lógica de predicados (clássica):

Definição 5.3.5. Seja Γ um conjunto de sentenças, e φ uma sentença. Dizemos que φ é *conseqüência lógica de* φ, denotado por $\Gamma \vDash \varphi$, se, para toda estrutura $\mathfrak{A}$, se $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Gamma$ então $\mathfrak{A} \vDash \varphi$. □

A noção de conseqüência lógica é a usual: uma sentença segue logicamente de um conjunto de premissas se, em toda interpretação em que as premissas são verdadeiras, a conclusão também é verdadeira.

Observação 5.3.6. Uma propriedade fundamental da lógica de predicados clássica (que também é satisfeita pela lógica proposicional clássica) é a chamada *compacidade*: se uma sentença φ segue logicamente de um conjunto Γ de premissas, então deve existir algum subconjunto finito Γ_0 de Γ tal que φ segue logicamente do conjunto Γ_0. □

Exercícios 5.3.7. Provar que $\neg(\forall x)\varphi$ e $(\exists x)\neg\varphi$ são logicamente equivalentes, isto é:

$$\neg(\forall x)\varphi \vDash (\exists x)\neg\varphi \quad \text{e} \quad (\exists x)\neg\varphi \vDash \neg(\forall x)\varphi.$$

Provar que $\neg(\exists x)\varphi$ e $(\forall x)\neg\varphi$ são logicamente equivalentes, isto é:

$$\neg(\exists x)\varphi \vDash (\forall x)\neg\varphi \quad \text{e} \quad (\forall x)\neg\varphi \vDash \neg(\exists x)\varphi.$$

Concluir que os quantificadores são interdefiníveis.

Vejamos a seguir alguns exemplos concretos aplicando os conceitos introduzidos.

Exemplo 5.3.8. Seja $\mathcal{S}$ uma assinatura que contém apenas uma constante c e um símbolo de predicado binário P. Considere uma $\mathcal{S}$-estrutura $\mathfrak{A} = \langle \mathbb{N}, I \rangle$ tal que $I(c) = 0$ e $I(P) = \{(n, m) \in \mathbb{N} \times \mathbb{N} \ : \ n < m\}$. Então $\mathcal{S}_{\mathfrak{A}}$ é a assinatura que contém as constantes c e c_n (para cada $n \in \mathbb{N}$), assim como o símbolo de predicado binário P. A $\mathcal{S}_{\mathfrak{A}}$-estrutura $\mathfrak{A}_{\mathbb{N}}$ é $\langle \mathbb{N}, I_{\mathbb{N}} \rangle$ tal que $I_{\mathbb{N}}(c) = I(c) = 0$; $I_{\mathbb{N}}(c_n) = n$ para cada $n \in \mathbb{N}$, e $I_{\mathbb{N}}(P) = I(P)$. Temos o seguinte:

1) $\mathfrak{A} \vDash (\forall x)(\exists y)P(x, y)$ sse $\mathfrak{A}_{\mathbb{N}} \vDash (\forall x)(\exists y)P(x, y)$ sse, para todo $n \in \mathbb{N}$, $\mathfrak{A}_{\mathbb{N}} \vDash (\exists y)P(c_n, y)$ sse, para todo $n \in \mathbb{N}$, existe $m \in \mathbb{N}$ tal que $\mathfrak{A}_{\mathbb{N}} \vDash P(c_n, c_m)$ sse, para todo $n \in \mathbb{N}$, existe $m \in \mathbb{N}$ tal que $(c_n^{\mathfrak{A}_{\mathbb{N}}}, c_m^{\mathfrak{A}_{\mathbb{N}}}) \in I(P)$ sse, para todo

$n \in \mathbb{N}$, existe $m \in \mathbb{N}$ tal que $(n, m) \in I(P)$ sse, para todo $n \in \mathbb{N}$, existe $m \in \mathbb{N}$ tal que $n < m$. Dado que esta última afirmação *concreta* pode ser constatada na estrutura $\mathfrak{A}$ como sendo verdadeira, podemos afirmar que $\mathfrak{A} \vDash (\forall x)(\exists y)P(x, y)$.

2) $\mathfrak{A} \vDash (\exists y)(\forall x)P(x, y)$ sse $\mathfrak{A}_{\mathbb{N}} \vDash (\exists y)(\forall x)P(x, y)$ sse existe $m \in \mathbb{N}$ tal que $\mathfrak{A}_{\mathbb{N}} \vDash (\forall x)P(x, c_m)$ sse existe $m \in \mathbb{N}$ tal que, para todo $n \in \mathbb{N}$, $\mathfrak{A}_{\mathbb{N}} \vDash P(c_n, c_m)$ sse existe $m \in \mathbb{N}$ tal que, para todo $n \in \mathbb{N}$, $(c_n^{\mathfrak{A}_{\mathbb{N}}}, c_m^{\mathfrak{A}_{\mathbb{N}}}) \in I(P)$ sse existe $m \in \mathbb{N}$ tal que, para todo $n \in \mathbb{N}$, $(n, m) \in I(P)$ sse existe $m \in \mathbb{N}$ tal que, para todo $n \in \mathbb{N}$, $n < m$. Dado que esta última afirmação é falsa na estrutura $\mathfrak{A}$, podemos afirmar que $\mathfrak{A} \nvDash (\exists y)(\forall x)P(x, y)$. □

Observando 1) e 2) no exemplo comprovamos formalmente que as sentenças $(\forall x)(\exists y)P(x, y)$ e $(\exists y)(\forall x)P(x, y)$ não são logicamente equivalentes. De fato, o que foi provado no exemplo anterior é que

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

pois existe uma estrutura em que a premissa é verdadeira mas a conclusão é falsa. Por outro lado, pode ser provado que

$$(\exists y)(\forall x)P(x, y) \vDash (\forall x)(\exists y)P(x, y).$$

Deixamos isto como exercicio para o leitor.

Exemplo 5.3.9. Seja $\delta := \alpha \wedge \beta \wedge \gamma$ tal que

$$\alpha := (\forall x)(\exists y)R(x, y); \quad \beta := \neg(\exists x)R(x, x);$$

$$\gamma := (\forall x)(\forall y)(\forall z)(R(x, y) \wedge R(y, z) \rightarrow R(x, z)).$$

Então δ é satisfatível, mas não é satisfatível em nenhum domínio finito.

Com efeito, para provar que δ é satisfatível, basta considerar a estutura $\langle \mathbb{N}, I \rangle$ tal que $I(R) = \{(n, m) \in \mathbb{N} \times \mathbb{N} \ : \ n < m\}$.

Suponhamos agora que $\mathfrak{A} = \langle D, I \rangle$ é uma estrutura tal que $\mathfrak{A} \vDash \delta$, logo $\mathfrak{A}_D \vDash \delta$, e então $\mathfrak{A}_D \vDash \alpha$, $\mathfrak{A}_D \vDash \beta$ e $\mathfrak{A}_D \vDash \gamma$. Observemos o seguinte:
(i) $\mathfrak{A}_D \vDash \alpha$ implica que, para todo $a \in D$, existe um $b \in D$ tal que $aI(R)b$. Podemos escolher, para cada $a \in D$, um elemento $f(a)$ de D tal que $aI(R)f(a)$. Desta maneira, obtemos uma função $f : D \rightarrow D$ tal que

$$aI(R)f(a) \quad \text{para todo } a \in D \qquad (1)$$

(ii) $\mathfrak{A}_D \vDash \beta$ implica que não existe $a \in D$ tal que $aI(R)a$. Logo,

$$a \neq f(a) \quad \text{para todo } a \in D \qquad (2)$$

(iii) $\mathfrak{A}_D \vDash \gamma$ implica que $I(R)$ é transitiva, isto é:

$$aI(R)b,\ bI(R)c \ \text{ implica } \ aI(R)c \ \text{ para todo } a, b, c \in D \qquad (3)$$

De (1) obtemos $aI(R)f(a)$ e $f(a)I(R)f(f(a))$, logo $aI(R)f(f(a))$, por (3), e então, por (1) e (2),

$$a \neq f(a),\ f(a) \neq f(f(a)),\ a \neq f(f(a)).$$

Isto é, obtemos três elementos diferentes a, $f(a)$, $f(f(a))$. Continuando com o raciocínio, obtemos uma seqüência infinita

$$a, f(a), f(f(a)), f(f(f(a))), \ldots$$

de elementos diferentes em D. Logo, D é infinito. □

Exemplo 5.3.10. Seja φ a sentença

$$(\forall x)(R(x) \vee S(x)) \rightarrow ((\forall x)R(x) \vee (\forall x)S(x)).$$

Vamos exibir uma estrutura $\mathfrak{A}$ tal que φ seja verdadeira, e uma estrutura $\mathfrak{B}$ tal que φ seja falsa.

Para a primeira parte, observe que $\mathfrak{A} \vDash \varphi$ sse $\mathfrak{A} \nvDash (\forall x)(R(x) \vee S(x))$ ou $\mathfrak{A} \vDash ((\forall x)R(x) \vee (\forall x)S(x))$ sse $\mathfrak{A} \nvDash (\forall x)(R(x) \vee S(x))$ ou $\mathfrak{A} \vDash (\forall x)R(x)$ ou $\mathfrak{A} \vDash (\forall x)S(x)$. Desta maneira, basta dar uma estrutura $\mathfrak{A} = \langle D, I\rangle$ satisfazendo alguma das seguintes propriedades:

- $I(R) = D$, logo $\mathfrak{A} \vDash (\forall x)R(x)$; ou
- $I(S) = D$, logo $\mathfrak{A} \vDash (\forall x)S(x)$; ou
- existe $a \in D$ tal que $\mathfrak{A}_D \nvDash (R(c_a) \vee S(c_a))$, isto é, existe $a \in D$ tal que $\mathfrak{A}_D \nvDash R(c_a)$ e $\mathfrak{A}_D \nvDash S(c_a)$, isto é, existe $a \in D$ tal que $a \notin I(R)$ e $a \notin I(S)$.

Para a segunda parte, observe que $\mathfrak{B} \nvDash \varphi$ sse $\mathfrak{B} \vDash (\forall x)(R(x) \vee S(x))$ e $\mathfrak{B} \nvDash ((\forall x)R(x) \vee (\forall x)S(x))$ sse $\mathfrak{B} \vDash (\forall x)(R(x) \vee S(x))$ e $\mathfrak{B} \nvDash (\forall x)R(x)$ e $\mathfrak{B} \nvDash (\forall x)S(x)$ sse
(i) $\mathfrak{B}_D \vDash (R(c_a) \vee S(c_a))$ para todo $a \in D$;

(ii) $\mathfrak{B}_D \nVdash R(c_a)$ para algum $a \in D$;
(iii) $\mathfrak{B}_D \nVdash S(c_a)$ para algum $a \in D$.

De (i) inferimos que $I(R) \cup I(S) = D$. De (ii) inferimos que existe $a \in D$ tal que $a \notin I(R)$. De (iii) inferimos que existe $b \in D$ tal que $b \notin I(S)$. Podemos, portanto, definir $\mathfrak{B} = \langle D, I \rangle$ tal que

$$D = \{a, b, c, d\}; \quad I(R) = \{a, b, d\}; \quad I(S) = \{c, d\}.$$

É fácil conferir que $\mathfrak{B} \nVdash \varphi$, pois as condições (i), (ii) e (iii) são satisfeitas. □

Exemplo 5.3.11. Consideremos novamente a assinatura de primeira ordem da aritmética $\mathcal{S}_{ar}$, introduzida no Exemplo 5.2.4. Seja $\bar{n}$ o termo $\overbrace{s(\ldots s(}^{n \text{ vezes}}\mathtt{0})\ldots)$, para todo $n > 0$ e definamos, por convenção, $\bar{0} := \mathtt{0}$. Lembremos que a relação binária de divisibilidade, "x divide a y" é dada pela fórmula

$$\mathtt{divide}(x, y) := (\exists z)(x \odot z \approx y).$$

O conceito de número par é facilmente definível a partir desta fórmula:

$$\mathtt{par}(x) := \mathtt{divide}(\bar{2}, x).$$

Isto é,

$$\mathtt{par}(x) := (\exists z)(\bar{2} \odot z \approx x).$$

Usaremos agora a fórmula $\mathtt{divide}(x, y)$ para definir em $\mathcal{S}_{ar}$ um conceito fundamental da aritmética: o conceito de número primo.

Lembremos que um número natural p é *primo* se $p \neq 1$ e p admite apenas divisores triviais (isto é, os únicos divisores de p são 1 e p). Usando a nossa assinatura podemos definir o seguinte predicado complexo:

$$\mathtt{primo}(x) := \neg(x \approx \bar{1}) \wedge (\forall y)(\mathtt{divide}(y, x) \to ((y \approx \bar{1}) \vee (y \approx x))).$$

A conhecida propredade da aritmética, "2 é o único primo par" pode ser expressada em $\mathcal{S}_{ar}$ pela sentença

$$\mathtt{primo}(\bar{2}) \wedge \mathtt{par}(\bar{2}) \wedge (\forall x)(\mathtt{primo}(x) \wedge \mathtt{par}(x) \ \to \ (x \approx \bar{2})).$$

É fácil provar que estas noções aritméticas coincidem com as noções "verdadeiras" da aritmética usual. Isto é, se consideramos a estrutura $\mathfrak{N} = \langle \mathbb{N}, I \rangle$ tal que $I(\mathtt{0}) = 0$, $I(s) = S$, $I(\oplus) = +$, $I(\odot) = \cdot$,

$$I(\preceq) = \{(n, m) \ : \ n \leq m\}, \ \text{ e } \ I(\approx) = \{(n, m) \ : \ n = m\}$$

(veja o Exemplo 5.2.4), então $\mathfrak{N} \vDash \texttt{divide}(\bar{n}, \bar{m})$ sse n divide a m. Logo $\mathfrak{N} \vDash \texttt{par}(\bar{n})$ sse n é par, e $\mathfrak{N} \vDash \texttt{primo}(\bar{n})$ sse n é primo. Mais ainda, para toda sentença ψ de $\mathcal{S}_{ar}$, $\mathfrak{N} \vDash \psi$ sse a correspondente sentença da aritmética "concreta" é verdadeira. Observe que éste é um (raro) caso em que os termos da linguagem expressam *todos* os elementos do domínio de uma estrutura dada (no caso, $\mathfrak{N}$). Com efeito, o número n é expressado pelo termo fechado $\bar{n}$, e então neste caso podemos prescindir do uso das constantes c_n (para $n \in \mathbb{N}$). □

Mencionaremos a seguir algumas propriedades da relação de conseqüência lógica $\vDash$ introduzida na Definição 5.3.5. Por simplicidade, a partir de agora escreveremos $\vDash \varphi$ no lugar de $\emptyset \vDash \varphi$.

Metateorema 5.3.12. *A relação de conseqüência lógica $\vDash$ satisfaz as seguintes propriedades:*
(1) $\vDash \varphi$ sse $\mathfrak{A} \vDash \varphi$ para toda estrutura $\mathfrak{A}$ (sse φ é válida).
(2) $\Gamma \vDash \varphi$ e $\Delta, \varphi \vDash \psi$ implica $\Gamma, \Delta \vDash \psi$ (propriedade do Corte*).*
(3) $\Gamma \vDash \varphi$ sse existe $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \vDash \varphi$ (Teorema da Compacidade).
(4) $\Gamma \vDash \varphi$ implica $\Gamma, \Delta \vDash \varphi$ para todo conjunto de sentenças Δ.
(5) Se φ é válida *(isto é, $\vDash \varphi$) então $\Gamma \vDash \varphi$ para todo Γ.*
(6) Se φ é contraditória *(isto é, $\vDash \neg\varphi$) então $\varphi \vDash \psi$ para toda ψ. Em particular: $\varphi, \neg\varphi \vDash \psi$ para toda φ e para toda ψ. Portanto, $\Gamma, \varphi, \neg\varphi \vDash \psi$ para todo Γ, para toda φ e para toda ψ.*
(7) $\Gamma \vDash \varphi \to \psi$ see $\Gamma, \varphi \vDash \psi$ (Meta-teorema da Dedução).
(8) $\Gamma \vDash \varphi$ e $\Gamma \vDash \neg\varphi$ implica $\Gamma \vDash \psi$ para toda ψ. □

Exemplo 5.3.13. Sejam

$$\begin{aligned} \Gamma = &\quad \{(\forall x)(F(x) \vee \neg G(x)),\ G(c),\ (\exists x)F(x),\ \neg G(b)\}, \\ \Delta = &\quad \{(\forall x)(F(x) \to G(x)),\ G(c)\}. \end{aligned}$$

Provar que $\Gamma \vDash F(c)$ mas $\Delta \nvDash F(c)$.

Para a primeira parte, suponha que $\mathfrak{A} = \langle D, I\rangle$ é uma estrutura tal que $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Gamma$. Em particular $\mathfrak{A} \vDash (\forall x)(F(x) \vee \neg G(x))$, logo $\mathfrak{A}_D \vDash (F(c_a) \vee \neg G(c_a))$ para todo $a \in D$. Em particular,

$$\mathfrak{A}_D \vDash (F(c_b) \vee \neg G(c_b))$$

para $b = I(c)$, e então $\mathfrak{A}_D \vDash (F(c) \vee \neg G(c))$. [4] Daqui $\mathfrak{A}_D \vDash F(c)$ ou $\mathfrak{A}_D \vDash \neg G(c))$, isto é,

$$\mathfrak{A}_D \vDash F(c) \text{ ou } \mathfrak{A}_D \nvDash G(c)) \qquad (1)$$

Lembrando que $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Gamma$ então, em particular, $\mathfrak{A} \vDash G(c)$, isto é, $\mathfrak{A}_D \vDash G(c)$. Por (1) inferimos que $\mathfrak{A}_D \vDash F(c)$, logo $\mathfrak{A} \vDash F(c)$.

Acabamos portanto de demonstrar o seguinte: para toda estrutura $\mathfrak{A}$, se $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Gamma$ então $\mathfrak{A} \vDash F(c)$. Isto prova que $\Gamma \vDash F(c)$, pela Definição 5.3.5.

Para a segunda parte, isto é, para provar que $\Delta \nvDash F(c)$, a estratégia é diferente: agora devemos *exibir* uma estrutura *particular*, digamos $\mathfrak{A}$, tal que $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Delta$, porém $\mathfrak{A} \nvDash F(c)$. Para isso, vamos entender o que significa que uma estrutura $\mathfrak{A}$ satisfaz toda sentença de Δ, para saber que tipo de estrutura vamos propor.

Temos que, se $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Delta$ então, em particular, $\mathfrak{A} \vDash (\forall x)(F(x) \rightarrow G(x))$ e então $\mathfrak{A}_D \vDash (F(c_a) \rightarrow G(c_a))$ para todo $a \in D$. Isto significa que $I(F) \subseteq I(G)$ (analize esta afirmação!), portanto já temos uma informação importante sobre a estrutura que estamos procurando. Por outro lado, $\mathfrak{A} \vDash G(c)$ e então $I(c) \in I(G)$ (analize esta afirmação!). Finalmente, queremos que $\mathfrak{A} \nvDash F(c)$, e então $\mathfrak{A}$ deve ser tal que $I(c) \notin I(F)$. Encontrar um exemplo de estrutura satisfazendo estes requerimentos é muito fácil: Trata-se de fornecer um conjunto não-vazio D, um elemento a de D e dois conjuntos A e B tais que $A \subseteq B \subseteq D$, $a \in B$ e $a \notin A$. Desta maneira, tomando $\mathfrak{A} = \langle D, I\rangle$ tal que $I(F) = A$, $I(G) = B$ e $I(c) = a$ teremos o seguinte: $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Delta$, mas $\mathfrak{A} \nvDash F(c)$. A existência de uma tal estrutura nos garante que $\Delta \nvDash F(c)$. □

Exemplo 5.3.14. Provar que se R é uma relação simétrica e transitiva, então: se dois elementos dados estão relacionados com um terceiro, então os dois elementos dados estão relacionados com exatamente os mesmos elementos.

Esta propriedade geral sobre relações pode ser demonstrada utilizando a noção de conseqüência lógica. Com efeito: considere $\Gamma = \{\alpha,\ \beta\}$ tal que

$$\alpha := (\forall x)(\forall y)(R(x, y) \rightarrow R(y, x));$$

[4] Observe que c e c_b denotam, na estrutura $\mathfrak{A}_D$, o mesmo individuo. Logo, c e c_b satisfazem as mesmas propriedades em $\mathfrak{A}_D$.

$$\beta := (\forall x)(\forall y)(\forall z)(R(x,y) \wedge R(y,z) \to R(x,z)),$$

e seja

$$\varphi := (\forall x)(\forall y)((\exists u)(R(x,u) \wedge R(y,u)) \to (\forall z)(R(x,z) \Leftrightarrow R(y,z)))$$

(como é usual, $(\delta_1 \Leftrightarrow \delta_2)$ denota a fórmula $(\delta_1 \to \delta_2) \wedge (\delta_2 \to \delta_1)$, para todas as fórmulas δ_1 e δ_2). Assim, a propriedade que queremos provar equivale a provar que $\Gamma \vDash \varphi$.

Seja $\mathfrak{A} = \langle D, I\rangle$ é uma estrutura tal que $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Gamma$. Logo, $I(R)$ é uma relação simétrica e transitiva (confira!). Por outro lado, $\mathfrak{A} \vDash \varphi$ sse $\mathfrak{A}_D \vDash \varphi$ sse

$$\mathfrak{A}_D \vDash ((\exists u)(R(c_a,u) \wedge R(c_b,u)) \to (\forall z)(R(c_a,z) \Leftrightarrow R(c_b,z))) \quad (*)$$

para todo $a, b \in D$. Assim, fixemos $a, b \in D$, e provemos que $(*)$ é o caso. Temos dois casos para analisar:
(1) $\mathfrak{A}_D \nvDash (\exists u)(R(c_a,u) \wedge R(c_b,u))$. Neste caso, trivialmente vale $(*)$.
(2) $\mathfrak{A}_D \vDash (\exists u)(R(c_a,u) \wedge R(c_b,u))$. Logo, existe $d \in D$ tal que $\mathfrak{A}_D \vDash (R(c_a,c_d) \wedge R(c_b,c_d))$, portanto $\mathfrak{A}_D \vDash R(c_a,c_d)$ e $\mathfrak{A}_D \vDash R(c_b,c_d)$. Daqui

$$(I_D(c_a), I_D(c_d)) \in I(R) \text{ e } (I_D(c_b), I_D(c_d)) \in I(R),$$

isto é,

$$(a,d) \in I(R) \text{ e } (b,d) \in I(R).$$

Dado que $I(R)$ é simétrica, de $(b,d) \in I(R)$ inferimos que $(d,b) \in I(R)$ e então, usando que $(a,d) \in I(R)$ e que $I(R)$ é transitiva, obtemos que $(a,b) \in I(R)$. Da simetria de $I(R)$, ainda obtemos que $(b,a) \in I(R)$.

Agora bem, para provar que vale $(*)$ deveríamos provar que

$$\mathfrak{A}_D \vDash (\forall z)(R(c_a,z) \Leftrightarrow R(c_b,z)).$$

Observe que $\mathfrak{A}_D \vDash (\forall z)(R(c_a,z) \Leftrightarrow R(c_b,z))$ sse

$$\mathfrak{A}_D \vDash (R(c_a,c_e) \Leftrightarrow R(c_b,c_e))$$

para todo $e \in D$ sse, para todo $e \in D$,

$$(a,e) \in I(R) \text{ sse } (b,e) \in I(R) \quad (**)$$

Seja $e \in D$. Se $(a,e) \in I(R)$ então, dado que $(b,a) \in I(R)$ e que $I(R)$ é transitiva, obtemos que $(b,e) \in I(R)$. Analogamente, se $(b,e) \in I(R)$ então, dado que que $(a,b) \in I(R)$, inferimos que $(a,e) \in I(R)$. Isto mostra $(**)$

e então $\mathfrak{A}_D \vDash (\forall z)(R(c_a, z) \Leftrightarrow R(c_b, z))$, ou seja, provamos que neste caso também vale $(*)$.

Dado que este argumento pode ser feito para todo $a, b \in D$, inferimos então que $\mathfrak{A} \vDash \varphi$. Noutras palavras, acabamos de demonstrar que, para toda estrutura $\mathfrak{A}$, se $\mathfrak{A} \vDash \gamma$ para toda $\gamma \in \Gamma$ então $\mathfrak{A} \vDash \varphi$. Isto significa que $\Gamma \vDash \varphi$ e a afirmação que queriamos demonstrar é, de fato, verdadeira. □

5.4 Um sistema axiomático para a lógica de predicados

Uma vez que definimos formalmente o conceito de linguagem de primeira ordem, assim como a sua semântica e a noção conseqüência lógica associada, o passo seguinte é definir a sua teoria da prova. Isto é, fornencer um método de prova de tipo *sintático*, em que a lógica de primeira ordem possa ser analisada apenas manipulando símbolos, sem interpretá-los, como é feito pelo método semântico introduzido na seção anterior. Vimos que, no caso da lógica proposicional, existem os seguintes métodos sintáticos de prova: sistemas axiomáticos, de sequentes, de dedução natural e de tablôs. Todos estes métodos podem ser estendidos para lógica de primeira ordem. Isto é, dado que a lógica de primeira ordem *estende* a lógica proposicional, é razoável que os métodos sintácticos de prova da lógica proposicional possam ser estendidos a métodos de prova para a lógica de predicados, acrescentando regras específicas para manipular os quantificadores.

Para não prolongar demais a nossa discusão, estudaremos aqui apenas um sistema axiomático para a lógica de primeira ordem, e não estudaremos as extensões dos métodos de sequentes, tablós e dedução natural para a lógica de primeira ordem.

Para poder definir as regras que precisam ser acrescentadas no sistema axiomático da lógica proposicional dado anteriormente, devemos introduzir um conceito:

Definição 5.4.1. Seja φ uma fórmula, x uma variável, e t um termo. Seja $\varphi(t)$ a fórmula obtida de φ substituindo todas as ocorrências livres de x em φ pelo termo t. Dizemos que t *é livre para* x *em* φ se nenhuma variável y de t ocorre quantificada em $\varphi(t)$ (a menos que essas ocorrências quantificadas de y ja estivessem em φ).

A melhor maneira de entender a definição anterior é através de exemplos:

Exemplos 5.4.2.
(1) Seja $\varphi = Q(x, y) \vee (\forall y)P(y, x)$ e $t = f(y, z)$. Então t não é livre para x

em φ. De fato, temos que

$$\varphi(t) = Q(f(y,z),y) \vee (\forall y)P(y,f(y,z)).$$

A primeira (de esquerda à direita) substituição de x por t não é problemática: nenhuma variável de t registra uma nova quantificação. Por outro lado, a segunda substituicão de x por t é problemática: agora a variável y de t aparece quantificada. É verdade que y já aparecia quantificada em φ, mas $\varphi(t)$ registra uma ocorrência quantificada de y *a mais*.
(2) Seja $\varphi = Q(x,y) \vee (\forall y)P(y)$ e $t = f(y,z)$. Então t é livre para x em φ. Com efeito, observe que

$$\varphi(t) = Q(f(y,z),y) \vee (\forall y)P(y).$$

Se bem que a variável y de t ocorre ligada em $\varphi(t)$, essa ocorrência já estava presente em φ. Dado que $\varphi(t)$ não registra *novas* ocorrências ligadas nem de y nem de z (sendo que y e z são todas as variáveis que ocorrem em t), então t é livre para x em φ.
(3) Se x não ocorre livre em φ, então *qualquer* termo t é livre para x em φ.
(4) A variável x é livre para x em φ, para toda variável x e toda fórmula φ.
(5) Se nenhuma variável de t ocorre em φ, então t é livre para x em φ, para toda variável x e toda fórmula φ. □

Podemos agora definir a extensão do sistema axiomático da lógica proposicional para a lógica de predicados. Lembremos que o sistema axiomático da lógica proposicional está baseado em regras escritas numa linguagem proposicional que apenas utiliza os conectivos $\neg$ (negação) e $\vee$ (disjunção). Continuaremos então utilizando essa linguagem mais simples (dado que os outros conectivos podem ser definidos em termos destes dois). Mais ainda, dado que os quantificadores são interdefiníveis (veja o Exercício 5.3.7) então a linguagem de primeira ordem que consideraremos agora somente utiliza o quantificador universal $\forall$, dado que a fórmula $(\exists x)\varphi$ equivale a $\neg(\forall x)\neg\varphi$. Em resumo, a partir de agora consideraremos que as fórmulas de primeira ordem complexas serão construídas a partir das fórmulas atômicas utilizando apenas os conectivos $\neg$ e $\vee$, e o quantificador $\forall$.

Acrescentamos então ao sistema axiomático da lógica proposicional clássica as seguintes regras de inferência:

(generalização em x) $\dfrac{\varphi \vee \psi}{\varphi \vee (\forall x)\psi}$ se x não ocorre livre em φ

(instanciação de x) $\dfrac{\varphi \vee (\forall x)\psi}{\varphi \vee \psi(t)}$ $\dfrac{\text{se } x \text{ não ocorre livre em } \varphi}{\text{e } t \text{ é livre para } x \text{ em } \psi}$

Observe que, na realidade, estamos acrescentado infinitas regras de inferência: cada variável x determina duas regras de inferência (generalização e instanciação).

Como sempre, $\Gamma \vdash \varphi$ significa que existe uma derivação da fórmula φ no sistema axiomático, a partir do conjunto de fórmulas Γ.

O importante *meta-teorema da dedução*, válido para o sistema axiomático da lógica proposicional, ainda continua sendo válido no sistema da lógica de primeira ordem, mas com certas restrições:

Metateorema 5.4.3 (Meta-teorema da dedução)**.** *Seja Γ um conjunto de fórmulas, e sejam φ e ψ duas fórmulas. Se $\Gamma \vdash (\varphi \to \psi)$ então $\Gamma, \varphi \vdash \psi$.* [5] *Por outro lado, se $\Gamma, \varphi \vdash \psi$ então $\Gamma \vdash (\varphi \to \psi)$,* **desde que** *exista uma derivação Π de ψ a partir de $\Gamma \cup \{\varphi\}$ satisfazendo o seguinte: se a regra de generalização em x ou de instanciação de x é utilizada em Π, então x não ocorre livre em φ. Em particular, se φ é uma sentença então $\Gamma, \varphi \vdash \psi$ see $\Gamma \vdash (\varphi \to \psi)$.*

Exemplos 5.4.4.
(1) $\varphi \vdash (\forall x)\varphi$ para toda fórmula φ e para toda variável x. Com efeito, podemos construir a seguinte derivação:

1. φ (hip.)
2. $(\forall x)\varphi \vee \varphi$ (exp)
3. $(\forall x)\varphi \vee (\forall x)\varphi$ (gen. em x)
4. $(\forall x)\varphi$ (elim).

(2) $\vdash ((\forall x)\varphi \to \varphi(t))$, se t é livre para x em φ. Com efeito, seja $\psi = (\neg(\forall x)\varphi \vee (\forall x)\varphi)$, e considere a seguinte derivação:

1. $(\forall x)\varphi$ (hip.)
2. $\neg\psi \vee (\forall x)\varphi$ (exp)

[5] Lembre que agora $(\varphi \to \psi)$ é uma notação para a fórmula $(\neg\varphi \vee \psi)$.

3. $\neg\psi \vee \varphi(t)$ (inst. de x)
4. ψ (axioma)
5. $\varphi(t)$ (modus ponens).

Logo, $(\forall x)\varphi \vdash \varphi(t)$. Daqui, pelo meta-teorema da dedução inferimos que $\vdash ((\forall x)\varphi \to \varphi(t))$. □

Com estes resultados, é possível obter o seguinte teorema fundamental:

Metateorema 5.4.5 (Teorema de Completude fraca)**.** *Seja φ uma sentença. Então: $\vDash \varphi$ see $\vdash \varphi$.* □

Dado que a relação de conseqüência semântica $\vDash$ é compacta (veja a Observação 5.3.6) e em virtude de que $\vDash$ e $\vdash$ satisfazem o meta-teorema da dedução (veja a Proposição 5.3.12(7) e o Teorema 5.4.3) então obtemos como corolário a completude forte do sistema axiomático proposto:

Corolário 5.4.6 (Teorema de Completude forte)**.** *Seja Γ um conjunto de sentenças, e seja φ uma sentença. Então*

$$\Gamma \vDash \varphi \quad see \quad \Gamma \vdash \varphi.$$

□

Em virtude deste último resultado, vemos que a lógica de primeira ordem, da mesma maneira que acontece com a lógica proposicional, admite duas 'faces' ou duas 'manifestações': por um lado, pode ser vista como um método sintático de demonstração, em que a partir de certas premissas podem ser obtidas (mediante a relação $\vdash$) certas conclusões, apenas utilizando regras (sintáticas) de inferência. Esta pespectiva generaliza o método silogístico de Aristóteles, o qual parte de uma visão análoga (inferir sintáticamente a partir de certas regras dadas –os silogismos).

Por outro lado, podemos assumir uma perspectiva semântica, de maneira tal que interpretamos a noção de conseqüência lógica $\vDash$ como sendo uma relação de 'preservação da verdade': se as premissas são verdadeiras (satisfeitas numa estrutura) então a conseqüência também deve ser verdadeira (satisfeita nessa estrutura). A importância do Teorema da Completude Forte (Corolário 5.4.6) reside em estabelecer que as duas perspectivas coincidem: as conclusões (sentenças) que podem ser logicamente obtidas a partir de um conjunto de sentenças, seja pela utilização de um método sintático, seja

pela utilização de um método semântico, são as mesmas. Para colocá-lo em termos simples: as noções de conseqüência lógica sintática e semântica são duas faces da mesma moeda.

Exemplo 5.4.7. Voltemos ao sistema ZF de Teoria de Conjuntos que descrevemos informalmente na Seção 5.1. Consideremos uma assinatura com apenas dois símbolos de relação binária, $\approx$ (representando a igualdade) e ε (representando a pertinência). Alguns dos axiomas de ZF podem ser escritos nesta assinatura da maneira seguinte:

- $(\forall x)(\forall y)((x \approx y) \Leftrightarrow (\forall z)(z \,\varepsilon\, x \;\Leftrightarrow\; z \,\varepsilon\, y))$ (axioma da extensionalidade).
- $(\forall x)(\exists y)(\forall z)((z \,\varepsilon\, y) \Leftrightarrow (z \,\varepsilon\, x \;\wedge\; \varphi(z)))$ (axioma de separação).
- $(\forall x)(\forall y)(\exists z)(\forall u)((u \,\varepsilon\, z) \Leftrightarrow (u \approx x \;\vee\; u \approx y))$ (axioma do par não-ordenado).
- $(\forall x)(\exists y)(\forall z)((z \,\varepsilon\, y) \Leftrightarrow (\exists u)(u \,\varepsilon\, x \;\wedge\; z \,\varepsilon\, u))$ (axioma da união).

Deixamos como exercício para o leitor escrever nesta linguagem (acrescentando, por simplicidade, uma constante para o conjunto vazio, um símbolo de função unária para o conjunto unitário e símbolos de função binárias para as operações de união e interseção) os axiomas do conjunto das partes, da regularidade, e do infinito. □

Finalizamos o capítulo com uma observação muito importante sobre a lógica de primeira ordem. Vimos que a lógica proposicional é *decidível*, isto é, existe um *procedimento efetivo* (um *algoritmo*) para que, em finitos passos, possamos responder *em todos os casos* à seguinte pergunta: "a fórmula φ é uma tautologia?"

Para isso, basta construír, por exemplo, uma tabela de verdade e constatar se todas as linhas devolvem o valor de verdade 1. Ou então, podemos exibir uma demonstração em algum sistema formal (sistema axiomático, ou sequentes, ou dedução natural, ou tablôs). Ainda, caso φ não seja uma tautologia, podemos, por exemplo, exibir um tablô completo aberto para $\neg\varphi$. Todos estes procedimentos são realizados em tempo finito e, o mais importante, *sempre podem ser realizados* de maneira que podemos determinar em un tempo finito se uma dada fórmula φ é uma tautologia ou não. Infelizmente, ese não é o caso com a lógica de predicados:

A lógica de primeira ordem é indecidível.

Esta última afirmação significa o seguinte: dada uma sentença φ escrita numa linguagem de primeira ordem, então evidentemente ou φ é válida ou φ não é válida. Se φ é válida, então certamente poderemos encontrar uma demonstração realizada em finitos passos (por exemplo, um tablô fechado finito para $\neg\varphi$, ou uma demonstração de φ num sistema de prova). Noutras palavras, dada a pergunta "φ é uma sentença válida?", se a resposta for "sim" então essa resposta sempre poderá ser dada num tempo finito. O problema é quando a sentença φ não é válida, isto é, quando a resposta para a pergunta é "não".

Com efeito, se φ é uma sentença que não é válida, então uma demonstração deste fato requer a construção de uma estrutura em que φ é falsa, isto é, a construção de um modelo para a sentença $\neg\varphi$. [6] E aqui aparece o problema: existem sentenças que *somente* admitem modelos infinitos. Por exemplo, a sentença δ do Exemplo 5.3.9. Se alguém perguntasse se a sentença $\neg\delta$ é universalmente válida ou não, então nenhum algoritmo poderia dar a resposta correta ("não"). Por exemplo, pelo método de tablô obteriamos um ramo aberto terminado, testemunhando que δ é satisfatível (provando que $\neg\delta$ não é válida). Porém, este ramo seria necessariamente infinito. A constatação de que $\neg\delta$ não é válida só pode ser feita através de uma análise formal (como, por exemplo, foi feito no Exemplo 5.3.9).

Portanto, o imenso ganho em poder expressivo que obtivemos pelo uso de linguagens de primeira ordem no lugar de linguagens proposicionais tem um preço a ser pago: a lógica obtida é indecidível. Desta maneira, é impossível, por exemplo, programar um computador para que analise *todos* os argumentos escritos em linguagens de primeira ordem, a menos que a linguagem seja muito simples, por exemplo que somente utilize predicados unários e não utilize símbolos de função (é conhecido que esta linguagem simples de primeira ordem é decidível). Para qualquer algoritmo (programa de computador) possível, sempre vai existir um argumento não válido tal que o algoritmo não conseguirá demonstrar a invalidez do argumento.

[6] Outra solução seria *demonstrar* que um dado sistema de prova *não* consegue provar φ. Mas a construção de uma tal demonstração não é mais um algoritmo, isto é, um computador não poderia ser programado para tentar *demonstrar matematicamente* que o sistema de prova não consegue provar uma sentença dada (caso a sentença não seja, de fato, demonstrável). Observe que cada sentença indemonstrável dada requer uma demonstração matemática *diferente* da sua indemonstrabilidade.

Capítulo 6

Introdução à Teoria dos Silogismos

Nste capítulo apresentaremos a famosa Teoria dos Silogismos de Aristóteles, na linguagem da lógica de primeira ordem apresentada no Capítulo 5. Veremos que a lógica dos silogismos é, a rigor, um fragmento da lógica de predicados.

6.1 O que representam os silogismos na lógica contemporânea

A Teoria dos Silogismos de Aristóteles (*Sobre a Interpretação e Primeiros Analíticos*, núcleo da lógica tradicional ou antiga) constitui o primeiro sistema formal já proposto, apesar de que, modernamente, podemos interpretá-la como um fragmento da teoria da quantificação em lógica formal: a partir do que já vimos sobre a Lógica de Predicados, a Teoria dos Silogismos é apenas uma parte do cálculo de predicados monádicos, onde só se permitem sentenças a respeito de propriedades unárias, e mesmo assim sob certos formatos especiais, como veremos. Contudo, isso não desmerece nem diminui o interesse pela Teoria dos Silogismos, mas, se não devemos cometer nenhuma falácia histórica, medindo a contribuição aristotélica com as ferramentas de hoje, também temos a obrigação de saber quanto de Lógica existe para além do domínio da Teoria dos Silogismos.

Nos *Primeiros Analíticos* Aristóteles devotou-se aos dois principais problemas da epistemologia da lógica enquanto ciência formal: como mostrar que uma conclusão proposta de fato segue de premissas aceitas, que dessa maneira formalmente a implicam, e o problema inverso: como mostrar que

uma conclusão proposta *não* segue das premissas aceitas.

O que quer que se considere (ou não se considere) ao que a Filosofia diga respeito, pelo menos um ponto é comum acordo: praticar Filosofia não é (só) pensar, é justificar. A justificativa filosófica passa necessariamente pela argumentação rigorosa. Uma definição ampla de argumento, como já vimos, é a seguinte e que continua a valer aqui também:

> **Argumento:** Um argumento é uma coleção de afirmações, uma das quais se chama "conclusão" e cuja verdade procura-se estabelecer; a partir de outras afirmações chamadas "premissas", as quais pretendem conduzir à conclusão (ou apoiá-la, ou persuadir-nos da sua verdade).

Os silogismos aristotélicos constituem um particular método argumentativo, e seu sistema, em muitos pontos similar aos sistemas de lógica moderna, envolve três partes:

i) Um domínio limitado de proposições, expresso em uma linguagem formalizada;

ii) Um método formal de dedução que permite estabelecer validade de argumentos a partir de um número ilimitado de premissas;

iii) Um método geral de produzir contra-argumentos de forma a estabelecer invalidades.

No entanto, a noção de argumento de que as várias ciências necessitam é muito mais ampla do que este particular método, e deveria ser óbvio para todo filósofo que, ainda que os argumentos expressáveis em termos de silogismos possam ser úteis e interessantes para a justificativa filosófica, eles são insuficientes. No final do século XVIII Kant poderia dizer (como de fato disse) que a lógica desde Aristóteles não tinha tido necessidade de avançar em nada, embora certamente errasse ao considerar a lógica aristotélica fosse 'completa e perfeita'. Leibniz, entre 1670 e 1680, já pensava em desenvolver um tipo de cálculo algébrico-combinatório capaz de capturar as regras do silogismo aristotélico.

Parte desse objetivo só foi conseguido dois séculos mais tarde, com George Boole in 1847. Boole concluiu que o método silogístico era demasiado restritivo e de aplicação estreita. No meio século seguinte o método algébrico

de Boole foi estendido por vários outros, como por exemplo por Charles Pierce que estudou o problema de modificar o método algébrico de Boole de forma a incorporar relações e ampliar o uso de quantificadores. Considere por exemplo a afirmação: "para todo homem existe um outro que é seu pai". Este tipo de afirmação tem obviamente uma forma lógica muito mais complexa do que as afirmações categóricas aristotélicas do tipo "todo homem é mortal", e foge às regras do silogismo.

Finalmente, em 1879, Gottlob Frege publicou sua obra contendo um sistema lógico mais ou menos equivalente à moderna lógica de predicados de primeira ordem, que incorpora a teoria dos silogismos (ou pelo menos uma interpretação moderna dessa teoria) como um mero fragmento. Por isso, a lógica matemática contemporânea tende a ver os silogismos como completamente irrelevantes, enquanto alguns filósofos ainda insistem na fantasia de que a lógica dos silogismos é tudo quanto basta para formalizar argumentos filosóficos.

É perfeitamente possível formalizar a lógica dos silogismos do ponto de vista de uma *teoria de primeira ordem*, de tal maneira que seja possível reproduzir formalmente as próprias derivações aristotélicas, se aceitarmos como corretas certas interpretações e simplificações. Considerando o interesse histórico e o alto grau de simetria que envolve os raciocínios silogísticos, é muito interessante estudar a lógica dos silogismos dentro de um estudo da lógica formal.

Consideramos como intuitivamente claro, em termos pré-formais, o significado dos operadores lógicos (também chamados *expressões sincategoremáticas*):

$\exists$ (existe), $\forall$ (para todo), $\neg$ (não), $\wedge$ (e), $\vee$ (ou), $\rightarrow$ (se ... então ...)

e o uso de variáveis e símbolos de predicados.

6.1.1 A linguagem dos silogismos

(1) *Termos* (ou idéias).
Por exemplo: "homem" e "número"são termos comuns, enquanto "Sócrates" e "dois" são termos singulares.

(2) *Predicados*.
Por exemplo: "mortal", "par".

(3) Expressões chamadas *sincategoremáticas*:
"todo", "algum", "é", "são", "não".

(4) *Proposições* ou *juízos*.
Por exemplo: "Todo homem é mortal", "Sócrates é homem", "cinco é par".

('5) *Silogismos* ou *inferências*.
Por exemplo:

> Todo homem é mortal
> Sócrates é homem
> Portanto, Sócrates é mortal.

A Teoria dos Silogismos trata de:

(1) *Proposições categóricas* (no sentido de "incondicionais"):
afirmação ou negação de relações entre classes.
Por exemplo: "Todo homem é mortal", isto é, "todo elemento da classe homem é elemento da classe mortal".

(2) *Proposições singulares*:
afirmação ou negação de pertinência de elemento à classe.
Por exemplo: "Sócrates é mortal", "Este é um homem".

6.1.2 Os quatro tipos de proposições categóricas

(1) *Afirmação universal* (notação: **A**)
Exemplo: "Todos os S são P".

(2) *Negação universal* (notação: **E**)
Exemplo: "Nenhum S é P".

(3) *Afirmação particular* (notação: **I**)
Exemplo: "Alguns S são P"ou "Algum S é P".

(4) *Negação particular* (notação: **O**)
Exemplo: "Alguns S não são P"ou "Algum S não é P".

A notação anterior vem de **A***FF***I***RMO* e *N***E***G***O**.
Observe que **A**, **E**, **I**, **O** diferem em

- qualidade (afirmam ou negam), e em
- quantidade (são universais ou particulares).

Em notação formal, as proposições categóricas podem ser interpretadas como:

A : $\forall x(S(x) \rightarrow P(x))$
"Para todo x, se vale $S(x)$ então vale $P(x)$" ou "Todo elemento da classe S é da classe P", ou ainda "Todo S é P".

E : $\forall x(S(x) \rightarrow \neg P(x))$
"Para todo x, se vale $S(x)$ então não vale $P(x)$" ou "Todo elemento da classe S é da classe não-P", ou ainda "Nenhum S é P".

I : $\exists x(S(x) \wedge P(x))$
"Existe x, tal que vale $S(x)$ e vale $P(x)$" ou "Existe algum elemento da classe S que é da classe P", ou ainda "Algum S é P".

O : $\exists x(S(x) \wedge \neg P(x))$
"Existe x, tal que vale $S(x)$ e não vale $P(x)$" ou "Algum elemento da classe S é da classe não P", ou ainda "Algum S é não-P", ou se preferir "Algum S não é P".

Podemos considerar A, E, I, O como operadores que atuam sobre classes, e denotar as proposições (por exemplo, relativas a S e P) como SAP, SEP SIP SOP (os quais, para facilidade de escrita podemos às vezes simplificar por SAP, SEP, SIP e SOP).

3. Os Princípios da lógica de Aristóteles

Na lógica formal de Aristóteles (isto é, nos textos onde se leva em conta não os raciocínios indutivos, mas aqueles desprovidos de conteúdo específico) podemos destacar alguns princípios básicos:

1) Aristóteles indica claramente que num raciocínio lógico-formal a conclusão deve depender apenas das premissas;

2) refere-se, pela primeira vez, à forma lógica, i.e, conceitos gerais, que podem ser denotadas por letras (mais tarde chamadas variáveis, mas é interessante observar que a idéia de variáveis já estivesse presente em Aristóteles);

3) mostra que é possível reduzir os infinitos raciocínios silogísticos a um conjunto finito de formas (mais tarde chamadas regras).

Além disso, Aristóteles explicita em forma lógica certos princípios já usados muitos antes dele, mas que antes se referiam aos seres e objetos existentes em geral: · Princípio da identidade Todo conceito (ou juízo) deve ser igual a si mesmo. · Princípio da Não-Contradição Uma proposição não pode ser verdadeira e falsa ao mesmo tempo. · Princípio do Terceiro Excluído Uma proposição deve ser verdadeira ou falsa, não havendo terceira possibilidade.

Referências Bibliográficas

[1] E.W. Beth. *The Foundations of Mathematics*. North-Holland, 1959.

[2] 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.

[3] G. Gentzen. Untersuchungen über das logische schliessen. *Mathematische Zeitschrift*, 39:176–210/405–431, 1934.

[4] K.J.J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.

[5] S. Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica*, 1:5–32, 1934.

[6] E. Mendelson. *Introduction to Mathematical Logic*. International Thomson Publishing, 4^{th} edition, 1997.

[7] D. Prawitz. *Natural Deduction*. Almqvist and Wiksell, Stockholm, 1965.

[8] R.M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.