Dulce Helena Marques Fernandes da Costa

O Problema da Completude nas Semânticas Intuicionistas

Orientadores
Professor Doutor Imme van den Berg
Professor Doutor Augusto Franco de Oliveira

Esta dissertação não inclui as críticas e sugestões feitas pelo júri

Mestrado em Matemática Aplicada
Biénio 2001/2003
Universidade de Évora

Universidade de Évora

O Problema da Completude nas Semânticas Intuicionistas

Dulce Helena Marques Fernandes da Costa

Orientadores

Professor Doutor Imme van den Berg

Professor Doutor Augusto Franco de Oliveira

Esta dissertação não inclui as críticas e sugestões feitas pelo júri

Mestrado em Matemática Aplicada

Biénio 2001/2003

Mestrado em Matemática Aplicada

ERRATA

da dissertação

Pág. 17

A leitura da primeira linha da definição 1.3.1 deve ser: Os parâmetros e as constantes são termos.

A leitura da definição 1.3.2 deve ser: Um termo fechado é um termo onde não ocorrem parâmetros.

Pág. 26

A última frase da página deve ser eliminada do texto.

Pág. 28

A leitura da linha 14 deve ser: Os elementos de K designam-se por *nós* do modelo $\mathfrak{K}$. Observemos que D e $\vDash$ determinam, em cada nó k, uma estrutura clássica $\mathfrak{A}(k)$, cujo universo é $D(k)$ e cujas relações $P_{\mathfrak{A}(k)}$ são definidas da seguinte forma: se P é um símbolo predicativo n-ário da linguagem e $t_1...t_n \in D(k)$ então $(t_1...t_n) \in P_{\mathfrak{A}(k)}$ sse $\vDash P^n(t_1...t_n)$, considerando que as constantes são interpretadas pelos mesmos elementos em todos os domínios.

Pág. 29

Na demonstração de 2.2.4, onde se lê «Lema 1», deve ler-se Lema 2.2.3.

Na demonstração de 2.2.5, onde se lê «definição 1», deve ler-se definição 2.2.1.

Pág. 31

Nos exemplos 1 e 2, onde se lê «Lema 1», deve ler-se Lema 2.2.3 e onde se lê «Lema 2», deve ler-se Lema 2.2.4.

Pág. 38

A primeira parte da demonstração de 2.3.13, correspondente a 30 linhas, deve ter a seguinte leitura:

Se $\Gamma \nvdash \varphi$ então, pelo Lema da Saturação, existe Γ' C-Saturado tal que $\Gamma' \supset \Gamma$ e $\Gamma' \nvdash \varphi$. Seja $\mathcal{L}'$ a linguagem, com conjunto de constantes C, das sentenças de Γ'.

Consideremos uma sequência numerável C_1, ..., C_n, ... de conjuntos disjuntos e numeráveis de constantes que não pertencem a $\mathcal{L}'$.

Tomemos um modelo de Kripke $\mathfrak{K} \equiv (K, \leq, D, \vDash)$ tal que:

(a) K é o conjunto de <u>todas as sequências finitas</u> de números naturais, incluindo a sequência vazia, consideradas na sua ordem natural. Assim, se $\vec{n}$ e $\vec{m}$ pertencem a K e $\vec{n} = \langle n_1, n_2, ...n_j \rangle$ e $\vec{m} = \langle m_1, m_2, ..., m_j, ...m_l \rangle$ então $\vec{n} \leq \vec{m}$ sse $j \leq l$ e $m_1 = n_1, m_2 = n_2, ..., m_j = n_j$, ou seja, o segmento inicial de $\vec{m}$, com comprimento j, é $\vec{n}$.

Refira-se que o primeiro nó do modelo é a sequência vazia, que denotaremos por $\langle\,\rangle$.

(b) Para cada $\vec{n}$ de comprimento $j > 0$, seja $C(\vec{n}) = C_0 \cup C_1 \cup ... \cup C_j$, onde $C_0 = C$.

O conjunto de sentenças válidas no primeiro nó é Γ' e o conjunto de constantes correspondente $C(\langle\,\rangle)$ é $C_0 = C$. Tomemos:

$D(\vec{n}) : = C(\vec{n})$ e $\mathcal{L}(\vec{n}) : = \mathcal{L}' \cup C(\vec{n})$

(c) Seja $At(\vec{n})$ o conjunto de todas as fórmulas atómicas na linguagem $\mathcal{L}(\vec{n})$. Vamos definir o conjunto de fórmulas atómicas válidas em cada nó, $\Sigma(\vec{n})$, por indução no <u>comprimento</u> j de $\vec{n}$.

Se $j = 0$, $\vec{n}$ é a sequência vazia. Relembremos que o conjunto de sentenças válidas neste primeiro nó é Γ' C-Saturado. Tomemos então $\Sigma(\langle\,\rangle) = \Gamma' \cap At(\langle\,\rangle)$.

Suponhamos que $\Sigma(\vec{n})$ está definido e vejamos como definir o conjunto de fórmulas atómicas válidas para cada um dos nós (de comprimento $j + 1$) acessíveis a partir de $\vec{n}$, por forma a ser saturado com respeito a uma determinada linguagem.

Consideremos uma enumeração de pares de sentenças em $\mathcal{L}(\vec{n})$, digamos $(\sigma_0, \tau_0), (\sigma_1, \tau_1), ..., (\sigma_i, \tau_i), ...$, tal que para cada $i \geq 0$, $\Sigma(\vec{n}) \cup \sigma_i \nvdash \tau_i$.

Aplicando o Lema da Saturação a $\Sigma(\vec{n}) \cup \sigma_i$ e τ_i, para cada i, obtemos $\Gamma(\vec{n}, i)$ saturado com respeito a uma nova linguagem $\mathcal{L}(\vec{n}, i)$, verificando-se que $\sigma_i \in \Gamma(\vec{n}, i)$ mas $\Gamma(\vec{n}, i) \nvdash \tau_i$. Seja $\Sigma(\langle n_1, n_2, ...n_j, i\rangle) : = \Gamma(\vec{n}, i) \cap At(\vec{n}, i)$.

Provemos agora o seguinte facto:

Pág. 40

A finalização da demonstração de 2.3.13 (linhas 17-21) deve ter a seguinte leitura:

Ao longo desta demonstração construímos um modelo $\mathfrak{K}$ cujo primeiro nó $\langle\,\rangle$, força Γ, visto que $\langle\,\rangle \vDash \Gamma'$ e $\Gamma \subset \Gamma'$.

Por outro lado, $\Gamma' \nvdash \varphi$ implica que $\Gamma(\langle\,\rangle) \nvdash \varphi$. Portanto, pelo lema que acabámos de provar, $\langle\,\rangle \nvDash \varphi$. Assim, tomando $k_0 = \langle\,\rangle$, temos que $k_0 \vDash \Gamma$ mas $k_0 \nvDash \varphi$.

Pág. 42

Na demonstração de 2.3.17,

onde se lê:

Seja $\Gamma(k)$ o conjunto de sentenças verdadeiras no nó $k \in K$.

Tomemos $\Gamma(k) = \begin{cases} \Gamma_1(k), \text{ se } k \in K_1 \\ \Gamma_2(k), \text{ se } k \in K_2 \\ C, \text{ se } k = k_0 \end{cases}$

Onde C é o conjunto de constantes da linguagem $\mathcal{L}$.

deve ler-se:

Para cada nó $k \in K$, definimos a estrutura associada:

$$\mathfrak{A}(k) = \begin{cases} \mathfrak{A}_1(k), \text{ se } k \in K_1 \\ \mathfrak{A}_2(k), \text{ se } k \in K_2 \\ |\mathfrak{A}|, \text{ se } k = k_0 \end{cases}$$

Onde $|\mathfrak{A}|$ é o conjunto de constantes da linguagem $\mathcal{L}$, se existir alguma, caso contrário $|\mathfrak{A}|$ contém um único elemento, digamos a. Para $i = 1, 2$ definimos os mergulhos $f_i : \mathfrak{A}(k_0) \hookrightarrow \mathfrak{A}(k_i)$ por $c \mapsto c_{\mathfrak{A}(k_i)}$ no caso de haver constantes em $\mathcal{L}$, ou, caso contrário, escolhemos arbitrariamente $a_1 \in \mathfrak{A}(k_1)$ e $a_2 \in \mathfrak{A}(k_2)$ e definimos $f_i(a) = a_i$.

Pág. 42

Na demonstração de 2.3.19, onde se lê «$k_c \not\Vdash \varphi(c)$», deve ler-se $k_c \not\Vdash \varphi(c)$ e onde se lê «$k_a \not\Vdash \varphi(a)$», deve ler-se $k_a \not\Vdash \varphi(a)$.

Pág. 43

Na identificação da linguagem de $\mathcal{LP}$, onde se lê «símbolo 0-ário !», deve ler-se símbolo unário !.

Pág. 51

Na linha 16, onde se lê AP$_6$. $\forall x(x \times y' = x \times y + x)$, deve ler-se AP$_6$. $\forall xy(x \times y' = x \times y + x)$.

Pág. 69

Na demonstração de 3.5.8, onde se lê «Proposição 1», deve ler-se Proposição 3.5.7 e onde se lê «Teorema 1 da secção *3.2*», deve ler-se Teorema 3.2.12.

Pág. 76

Na linha 30, onde se lê «nota de rodapé da página 43», deve ler-se nota de rodapé da página 44.

Pág. 77

Na nota n.º 32, onde se lê «nota de rodapé n.º18», deve ler-se nota de rodapé da página 54.

Na nota n.º 33, onde se lê «nota de rodapé n.º13», deve ler-se nota de rodapé da página 44.

Pág. 84

Na demonstração de 3.5.20, onde se lê «Lema 3.5.13», deve ler-se Lema 3.5.12.

Pág. 85

Na demonstração de 3.5.22, onde se lê «Corolário 3.4.1», deve ler-se Corolário 3.4.2.

Pág. 86

Na definição 3.6.1, onde se lê «translação», deve ler-se tradução.

Agradecimentos

Ao meu orientador Imme Van den Berg, por ter aceite direccionar a minha dissertação, pela confiança que depositou em mim e neste trabalho, por sempre me encorajar a prosseguir e pela disponibilidade constante em discutir questões relacionadas com a minha investigação.

Ao meu co-orientador Augusto Franco de Oliveira, pelo contributo dado e pela forma como, com o seu entusiasmo nas aulas de Lógica e Fundamentos de Geometria na FCUL e na apresentação do Curso Introdutório à Lógica Intuicionista na UE, estimulou o meu gosto pela Lógica e despertou o meu desejo inicial de investigar nesta área.

A todos os meus colegas do biénio 2001/2003 do Mestrado em Matemática Aplicada da UE, pela amizade e partilha de ideias ao longo do ano curricular.

Ao meu professor de Matemática do ensino básico e secundário, Alcindo Martinho, que desde cedo me incutiu o gosto pelo rigor e simplicidade na abordagem de um problema.

Ao Luís por todo o apoio emocional e pelo respeito que demonstrou ter pelas minhas opções enquanto desenvolvi este trabalho.

Finalmente agradeço aos meus pais José e Ilusinda e aos meus irmãos Rui, Cristina e Susana, pelos sábios conselhos, pelo incentivo permanente e pelo carinho com que acolhem as minhas decisões. A toda a minha família pelo apoio que sempre me dedicou.

Índice

Sumário

Uma interpretação informal da lógica intuicionista foi traduzida nos anos 30 do séc.XX pela semântica Brouwer-Heyting-Kolmogorov, que explicita *verdade intuicionista* como *demonstrabilidade* (num sentido intuitivo).

Desde então surgiram várias semânticas *adequadas* (mais ou menos formais) para esta lógica, nomeadamente a semântica de Kripke. Esta semântica reflecte uma abordagem em termos de valuações, mais dinâmica do que a clássica. Ao admitir a existência de diferentes estádios de conhecimento (possíveis mundos), os modelos de Kripke permitem interpretar "falso" como "ainda não é verdadeiro". Sendo uma semântica relativamente simples permite, no entanto, um conjunto diversificado de convenientes aplicações.

Contudo, o problema de formalizar a semântica BHK e estabelecer a completude da lógica proposicional intuicionista com respeito a esta semântica, ficou em aberto durante décadas. Apresentamos um dos mais recentes avanços nesta matéria: a semântica de Artemov. Demonstramos a completude da sua lógica de demonstrabilidade explícita, $\mathcal{LP}$, que além de fornecer uma resposta ao problema, permite simultaneamente encontrar uma semântica de demonstrabilidade adequada para a lógica modal S4 de Gödel.

Abstract

The problem of establishing completeness in Intuitionistic Semantics

An intended informal meaning of intuitionistic logic was given in the 1930s by the Brouwer-Heyting-Kolmogorov semantics which understands *intuitionistic truth* as *provability*.

Since then, a number of (more or less formalized) adequate semantics, such as Kripke's semantics, appeared for intuitionistic logic. This particular semantics reflects a more dynamic approach then that of classical assignment of truth values to the propositional variables. Admiting various stages of knowledge (possible worlds) Kripke models interpret the value "false" as "not yet true". It is a fairly simple semantics but it is convenient for its applications.

The problem of formalizing the BHK semantics and establishing the completeness of propositional intuitionistic logic with respect to this semantics remained open for decades. We present one of the most recent advances in this area: Artemov's semantics. We establish the completeness of his logic of explicit provability ($\mathcal{LP}$) which provides an answer to the problem and an adequate provability semantics for modal logic S4 suggested by Gödel.

«It does not make sense to think of truth or falsity of a mathematical statement independently of our knowledge concerning the statement. A statement is true if we have a proof of it, and false if we can show that the assumption that there is a proof for the statement leads to a contradiction. For an arbitrary statement we can therefore not assert that it is either true or false.»

A. S. Troelstra e D. van Dalen, *Constructivism in Mathematics. An Introduction*, vol.1 pág.4 (1988)

Introdução

A abordagem construtiva da Matemática, segundo o espírito de Brouwer (1881-1966) e de Heyting (1898-1980) ficou conhecida como *Intuicionismo*.

Os argumentos filosóficos de Brouwer baseiam-se essencialmente na intuição matemática. Um dos aspectos mais relevantes do seu trabalho é requerer a "existência efectiva", ou seja, a obrigatoriedade de especificação dos objectos matemáticos que afirmamos existirem.

A origem da diferença entre a lógica intuicionista e a lógica clássica reside, portanto, no significado de *verdade*. De acordo com Brouwer, verdade significa demonstrabilidade. Esta noção de verdade reflecte-se essencialmente na interpretação dada ao conectivo $\vee$ e ao quantificador $\exists$.

A interpretação intuicionista dos operadores lógicos, fornecida implícita e informalmente por Brouwer em 1908, foi introduzida com maior precisão e de forma explícita, por Heyting em 1931 e por Kolmogorov em 1932, com abordagens aparentemente distintas mas que acabaram por se fundir numa única interpretação que ficou conhecida como interpretação BHK (Brouwer-Heyting-Kolmogorov). Nesta semântica da lógica intuicionista, as noções de prova construtiva e operação construtiva surgem como noções primitivas e portanto bastante vagas. Talvez por esse motivo, BHK nunca se tenha tornado um instrumento versátil como a semântica clássica.

Das diferentes semânticas que desde então surgiram para a lógica intuicionista, escolhemos, como âmbito deste trabalho, a semântica de Kripke e a semântica de Artemov. A semântica de Kripke, fornece uma técnica rápida para verificação da não derivabilidade de determinados resultados, nomeadamente fórmulas deriváveis na lógica clássica que o não são na lógica intuicionista. A semântica de Artemov, por sua vez, vai ao encontro dos ideais construtivistas, na medida em que formaliza a interpretação BHK revelando-se adequada à lógica intuicionista.

No capítulo introdutório, depois de uma breve apresentação do raciocínio construtivista, descrevemos a interpretação BHK e apresentamos um sistema de dedução natural possível para a lógica intuicionista.

No segundo capítulo desenvolvemos a semântica de Kripke que, na nossa opinião, é referência obrigatória num trabalho alusivo à lógica intuicionista, uma vez que a definição de modelo de Kripke traduz a forma como um construtivista concebe a evolução do conhecimento matemático. Na secção 2.3 deste capítulo provamos que a lógica intuicionista é válida e completa com respeito a esta semântica.

Finalmente, no capítulo 3, exploramos o sistema $\mathcal{LP}$, introduzido por Sergei Artemov em 1995. Este sistema vem responder a uma questão que permaneceu em aberto durante décadas: encontrar uma semântica formal para a lógica proposicional intuicionista, capaz de traduzir verdade intuicionista como demonstrabilidade e especificar os conectivos através de operações com provas, trabalho realizado informalmente pela semântica BHK. Propomo-nos demonstrar a adequação de $\mathcal{LP}$, utilizando uma interpretação aritmética da sua linguagem, na teoria **AP**. Na secção 3.6 deste último capítulo faremos um esboço do trabalho que seria necessário desenvolver para obter a formalização de BHK e a completude da lógica proposicional intuicionista com respeito a essa semântica.

Capítulo 1

Uma introdução à lógica intuicionista — semântica e sintaxe

1.1 O raciocínio/pensamento construtivista

A matemática tradicional, não construtivista, e a lógica que lhe está associada, designam-se por *clássicas*. Por outro lado, a matemática construtivista de Brouwer, e a correspondente lógica, dizem-se intuicionistas.

Brouwer rejeita a maneira clássica de fazer matemática, criticando muitos pontos tradicionalmente aceites. Considera acima de tudo que a matemática é uma actividade mental construtiva. Consequentemente, tanto os objectos matemáticos, como as suas propriedades, são construções *mentais* **(Dummet, 2000)**. Nesse sentido, a verdade não é absoluta. Um indivíduo não pode considerar que uma proposição é verdadeira ou falsa independentemente do seu conhecimento. Uma proposição é verdadeira para um indivíduo se ele tem uma construção (demonstração) que a estabelece. Segundo Brouwer, neste processo de decisão da validade da proposição, a linguagem não deve representar nenhum papel, sendo introduzida apenas por uma questão de comunicação.[1] Por esse motivo, e contrariamente à opinião de muitos matemáticos, para Brouwer a lógica não é precedente à matemática mas sim dependente dela.

Por outro lado, Brouwer considera que a lógica clássica não é 'fiável'. Um dos argumentos que apresenta é baseado no facto de fornecer demonstrações de existência de determinados objectos sem incluir na demonstração a construção dos referidos objectos. Para um intuicionista, a sentença "existe um número primo maior do que $10^{10^{10}}$" é mais forte do que a sentença "é impossível que o conjunto dos números primos se esgote antes do número $10^{10^{10}}$". Logo, para provar que a primeira é verdadeira não basta provar que a segunda o é. Para provar a primeira será necessário encontrar, no conjunto dos números primos, um elemento maior do que $10^{10^{10}}$ e exibi-lo.

Há duas leis, incorrectas segundo o pensamento construtivista, que são, frequentemente, responsáveis pelo tipo de demonstrações acima referidas, nomeadamente o *Princípio do terceiro excluído* e a *Eliminação da dupla negação* (utilizada no método de redução ao absurdo). Para nos convencermos da não validade destas leis clássicas, na lógica

[1]Brouwer, 1912: "And in the constuction of [all mathematical sets of units which are entitled to that name] neither the ordinary language nor any symbolic language can have any other rôle than that of serving as a nonmathematical auxiliary, to assist the mathematical memory or to enable different individuals to build up the same set." **(Epstein, 1995)**

intuicionista, consideremos os seguintes contra-exemplos baseados em questões que estão, até ao momento, sem resposta.

1. O Princípio do terceiro excluído não é válido.

A título de exemplo consideremos a conjectura, G, de Goldbach, segundo a qual cada número par maior do que 4 é a soma de dois números primos ímpares.

Poderá um intuicionista afirmar $G \vee \neg G$? Rapidamente encontramos exemplos particulares que nos levam a acreditar que a conjectura é verdadeira: $6 = 3 + 3$; $8 = 3 + 5$; $10 = 5 + 5$; $12 = 5 + 7$; $14 = 7 + 7$; $16 = 3 + 13$; ... No entanto, como não conseguimos efectuar uma pesquisa infinita percorrendo todos os números naturais pares maiores do que 4, com este processo poderemos apenas obter, eventualmente, um contra-exemplo, mas nunca uma demonstração da validade da conjectura. Por outro lado, como ainda não foi encontrado um número par maior do que 4 que não seja soma de dois números primos ímpares, não podemos provar $\neg G$. Para poder afirmar $G \vee \neg G$, um intuicionista teria que apresentar uma construção que decidisse qual das alternativas é verdadeira, demonstrando assim a validade de G ou a validade de $\neg G$.

2. A Eliminação da dupla negação não é válida.

Consideremos a dízima que representa o número real π: $3,14159265$.... Embora se conheçam mais de um trilião de casas decimais, poucas regularidades foram ainda encontradas nesta expansão decimal. Consideremos, por exemplo, a sentença φ: “Existe uma sequência de 9 noves na dízima correspondente a π”.[2]

Embora até ao momento não se tenha encontrado uma tal sequência de noves, e não haja, portanto, fundamento para afirmar a validade de φ, também não se conseguiu provar que essa sequência não existe. Sendo assim, não podemos, com o nosso conhecimento actual, afirmar ou refutar ϕ. No entanto, podemos justificar a validade de $\neg(\varphi \wedge \neg\varphi)$ já que as duas hipóteses em aberto não podem ser simultaneamente verdadeiras. Com um argumento análogo se justifica a validade de $\neg(\neg\varphi \wedge \neg\neg\varphi)$ que, como veremos adiante, também na lógica intuicionista é equivalente a $\neg\neg(\varphi \vee \neg\varphi)$. Logo $\neg\neg(\varphi \vee \neg\varphi)$ é intuicionisticamente válida, mas $(\varphi \vee \neg\varphi)$ não o é. Portanto, $\neg\neg\psi \rightarrow \psi$ não pode ser intuicionisticamente válida.

Os exemplos que acabámos de analisar sugerem já que o conectivo $\vee$ e o quantificador $\exists$ são de extrema importância no pensamento construtivista. Reparemos que o primeiro exige uma decisão efectiva e o segundo a exibição de um determinado objecto que procuramos e

[2]Uma sequência de oito noves é encontrada na posição 36 356 642 contada a partir da primeira casa decimal (www.joyofpi.com/pifacts.html).

suspeitamos que exista. Comparemos este tipo de raciocínio com o clássico. Classicamente podemos demonstrar que $\varphi \vee \psi$ é verdadeira provando que se tem $\neg(\neg\varphi \wedge \neg\psi)$, sem que se conheça qual das componentes φ ou ψ é demonstrável. Também $\exists x\varphi(x)$ pode ser demonstrada classicamente por redução ao absurdo, mostrando que a sua negação conduz a uma contradição, sem que fiquemos a conhecer um objecto apropriado a tal que $\varphi(a)$.

1.2 A interpretação BHK

Apresentemos então uma interpretação informal do conceito de "demonstração", conhecida por interpretação BHK (Brouwer-Heyting-Kolmogorov). Esta heurística foi inicialmente proposta por Heyting, que tentou ser fiel à filosofia de Brouwer e ao espírito construtivista. Ao introduzir a sua proposta de semântica de demonstrabilidade para a lógica intuicionista, Kolmogorov utilizou uma abordagem fundamentalmente distinta da apresentada por Heyting.

Enquanto Heyting interpretava a lógica intuicionista baseado na noção de construção e prova de um intuicionista, Kolmogorov (e mais tarde Gödel) esboçou uma interpretação desta lógica baseada na noção usual de prova (que, no sentido clássico, é solução de um problema). A proposta de Kolmogorov passava, portanto, por definir a lógica intuicionista através da matemática clássica.

No trabalho que iremos desenvolver ao longo do terceiro capítulo é essencial ter presente os objectivos de cada autor e distinguir as duas abordagens, no entanto, por agora, se encararmos cada proposição como um problema e a solução do problema como a respectiva prova, tal como Heyting posteriormente admitiu (**Troelstra** e **van Dalen**, **1988**), a essência das duas interpretações é a mesma.

BHK descreve o que é uma demonstração de uma fórmula composta a partir das demonstrações das suas componentes. Supomos que a noção primitiva de demonstração de uma fórmula atómica é conhecida (como já foi referido entende-se por demonstração um tipo particular de construção mental). Supomos também que as construções, que serão designadas pelas letras $A, B, C, \ldots$, têm determinadas propriedades de fecho. Por exemplo, se A e B são construções, então o par ordenado (A, B) é uma construção e $A(B)$ é uma construção que se obtém aplicando a construção A à construção B.

Consideremos como primitivos os conectivos $\wedge$, $\vee$, $\rightarrow$ e $\perp$.[3] Suponhamos que **D** é um conjunto não vazio de objectos já construídos que, por abuso, vamos identificar com as constantes que os designam.

---

[3]O símbolo $\perp$ denota o absurdo (contradição ou proposição sempre falsa). Se considerássemos $\neg$ um conectivo primitivo, em vez de $\perp$, teríamos que acrescentar à interpretação a suposição de que nenhuma construção demonstra uma contradição.

BHK estipula que:

1. A é uma demonstração de $\varphi \wedge \psi$ sse A é um par ordenado (B, C) tal que B é uma demonstração de φ e C é uma demonstração de ψ.

2. A é uma demonstração de $\varphi \vee \psi$ sse A é um par ordenado (a, C), em que $a \in \{0, 1\}$ e C é uma construção, tal que, se $a = 0$ então C demonstra φ, e se $a = 1$ então C demonstra ψ.

3. A é uma demonstração de $\varphi \to \psi$ sse A é uma construção que converte cada demonstração B de φ numa demonstração $A(B)$ de ψ.

4. Nenhuma construção é uma demonstração de $\perp$.

5. A é uma demonstração de $\forall x \varphi(x)$ sse A é uma construção tal que, para cada $d \in \mathbf{D}$, $A(d)$ é uma demonstração de $\varphi(d)$.

6. A é uma demonstração de $\exists x \varphi(x)$ sse A é um par ordenado (d, C) tal que $d \in \mathbf{D}$ e C é uma demonstração de $\varphi(d)$. ■

1.2.1 Observações:

1. O conectivo $\neg$ define-se por $\neg\varphi := \varphi \to \perp$. Resulta desta definição que A é uma demonstração de $\neg\varphi$ sse A é uma construção que converte cada demonstração B de φ numa demonstração $A(B)$ de $\perp$. Como esta última não existe, equivale a dizer que é impossível encontrar uma demonstração B de φ.

2. A interpretação BHK dá-nos uma ideia intuitiva do que é ou não correcto na lógica intuicionista. Reparemos, por exemplo, que esta interpretação dos conectivos é concordante com as propriedades da disjunção e da existência defendidas por Brouwer. A primeira propriedade diz respeito ao facto de Brouwer considerar que uma demonstração de $\varphi \vee \psi$ consiste ou numa demonstração de φ ou numa demonstração de ψ. A propriedade da existência traduz o facto de uma demonstração de $\exists x \varphi(x)$ consistir, obrigatoriamente, numa construção de um objecto apropriado, *d*, do domínio (ou na sua apresentação, caso o domínio seja suficientemente simples e não se levantem dúvidas acerca da pertença do elemento ao domínio) e numa demonstração de $\varphi(d)$.

Com os exemplos que se seguem ilustramos a utilização desta interpretação informal na justificação da validade (intuicionista) de algumas regras.

1.2.2 Exemplos:

1. $\varphi \wedge \psi \to \varphi$ é verdadeira, quaisquer que sejam φ e ψ.

Seja (A, B) uma demonstração de $\varphi \wedge \psi$. Tomemos uma construção C tal que $C(A, B) = A$ (aplicação habitualmente designada por primeira projecção). C é uma construção que converte cada demonstração de $\varphi \wedge \psi$ numa demonstração de φ. Logo, por definição, C é uma demonstração de $\varphi \wedge \psi \to \varphi$.

2. $(\neg\varphi \wedge \neg\psi) \rightarrow \neg(\varphi \vee \psi)$ é verdadeira, quaisquer que sejam φ e ψ.

Seja (A, B) uma demonstração de $\neg\varphi \wedge \neg\psi$. Pela definição do conectivo $\neg$, A converte uma demonstração de φ numa demonstração de $\perp$ e B converte uma demonstração de ψ numa demonstração de $\perp$. Pretendemos encontrar uma construção C que transforme (A, B) numa demonstração $C(A, B)$ de $\neg(\varphi \vee \psi)$. Tomemos uma demonstração (a, D) de $\varphi \vee \psi$.

Seja E uma construção tal que $E(a, D) :\, = A(D)$ se $a = 0$ e $E(a, D) :\, = B(D)$ se $a = 1$.

Definida deste modo, a contrução E transforma uma demonstração de $\varphi \vee \psi$ numa demonstração de $\perp$, portanto E demonstra $\neg(\varphi \vee \psi)$.

Em qualquer dos casos ficou explícita uma construção C que transforma a demonstração (A, B) de que partimos numa demonstração de $\neg(\varphi \vee \psi)$.

Sendo assim, definindo $C(A, B) = E$, concluimos que C demonstra $(\neg\varphi \wedge \neg\psi) \rightarrow \neg(\varphi \vee \psi)$.

3. $\neg(\varphi \vee \psi) \rightarrow (\neg\varphi \wedge \neg\psi)$ é verdadeira, quaisquer que sejam φ e ψ.

Seja A uma demonstração de $\neg(\varphi \vee \psi)$. Então A converte uma demonstração de $\varphi \vee \psi$ numa demonstração de $\perp$. Pretendemos encontrar uma construção B que transforme A numa demonstração $B(A)$ de $\neg\varphi \wedge \neg\psi$. Tomemos uma demonstração C de φ e uma demonstração D de ψ. Desse modo $(0, C)$ e $(1, D)$ são demonstrações de $\varphi \vee \psi$ e, portanto, $A(0, C)$ e $A(1, D)$ são demonstrações de $\perp$. Vamos definir uma construção E tal que $E(C) :\, = A(0, C)$ e uma construção F tal que $F(D) :\, = A(1, D)$. Então E transforma C numa demonstração de $\perp$ e F transforma D numa demonstração de $\perp$. Sendo assim, E demonstra $\neg\varphi$ e F demonstra $\neg\psi$. Logo a construção (E, F) demonstra $\neg\varphi \wedge \neg\psi$. Finalmente, definindo B tal que $B(A) :\, = (E, F)$ obtemos a construção pretendida, concluindo que B demonstra $\neg(\varphi \vee \psi) \rightarrow (\neg\varphi \wedge \neg\psi)$.

4. Para quaisquer φ e ψ, se $\varphi \leftrightarrow \psi$ é verdadeira então $\neg\varphi \leftrightarrow \neg\psi$ também é verdadeira.[4]

Basta demonstrar que se $\varphi \rightarrow \psi$ é verdadeira então $\neg\psi \rightarrow \neg\varphi$ também o é, uma vez que a demonstração da outra implicação seria análoga.

Tomemos uma demonstração E de $\varphi \rightarrow \psi$.

Seja A uma demonstração de $\neg\psi$ e B uma demonstração de φ. Então $C = E(B)$ demonstra ψ. Como A converte uma demonstração de ψ numa demonstração de $\perp$ concluímos que $A(C)$ demonstra $\perp$. Seja $D = A(C)$ e tomemos uma construção F tal que $F(B) = D$. Notemos que a construção F assim definida demonstra $\neg\varphi$.

Para finalizar, tomemos uma construção G tal que $G(A) = F$. G converte uma demonstração de $\neg\psi$ numa demonstração de $\neg\varphi$, logo demonstra $\neg\psi \rightarrow \neg\varphi$. ■

---

[4]O conectivo $\leftrightarrow$ define-se por $\varphi \leftrightarrow \psi :\, = (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

1.2.3 Observação:

Notemos que, fazendo $\psi = \neg\varphi$ nos exemplos **2.** e **3.**, obtemos a validade de $(\neg\varphi \wedge \neg\neg\varphi) \leftrightarrow \neg(\varphi \vee \neg\varphi)$. Sabendo que esta equivalência é válida, basta aplicar o resultado obtido no exemplo **4.** para obter a validade de $\neg(\neg\varphi \wedge \neg\neg\varphi) \leftrightarrow \neg\neg(\varphi \vee \neg\varphi)$, referida no exemplo 2. da secção *1.1*.

1.3 Sistema de dedução natural para a lógica intuicionista

A partir deste momento torna-se necessário um sistema formal, com um conjunto de regras lógicas, para que não tenhamos que recorrer à interpretação BHK sempre que se pretenda justificar a utilização de uma lei num determinado argumento construtivista. As regras escolhidas deverão traduzir, o mais fielmente possível, o significado dos conectivos e dos quantificadores, implícito na interpretação BHK.

O sistema de dedução natural que vamos descrever para a lógica intuicionista, obtém-se, essenciamente, do sistema de dedução natural utilizado na lógica clássica, através da substituição da regra $(\neg\neg^-)$, que não é intuicionisticamente válida, pela regra ($\perp$), tanto no caso da lógica proposicional como no caso da lógica de predicados.

É de referir que, analogamente ao que acontece na lógica clássica, poderíamos ter escolhido uma formalização através do cálculo de sequentes **(Mints, 2000)** ou uma formalização do tipo axiomático, normalmente designada por sistema de Hilbert **(Troelstra** e **van Dalen, 1988)**, em vez da formalização que apresentamos, cujo sistema não contém axiomas mas apenas regras (sistema devido a Gentzen).

Antes de apresentar o conjunto de regras de inferência em que se baseia o sistema dedutivo intuicionista vamos apresentar o alfabeto a utilizar na linguagem elementar ou de primeira ordem $\mathcal{L}$ **(Oliveira, 1996)**.

Adoptamos os seguintes símbolos lógicos:

- Conectivos proposicionais $\wedge$, $\vee$, $\perp$, $\rightarrow$; [5]
- Quantificadores $\forall, \exists$;
- Parênteses (,);
- Variáveis individuais x, y, z, ... (possivelmente com índices);
- Símbolo de igualdade $=$;
- Parâmetros α, β, γ, ...(possivelmente com índices)[6];

[5]O conectivo $\neg$ é definido como foi referido na secção *1.2* e o conectivo $\leftrightarrow$ é utilizado, como temos já vindo a fazer, para abreviar uma dupla implicação.

[6]Os parâmetros irão desempenhar o papel reservado às variáveis livres nas fórmulas (def. 1.3.5). Apenas as variáveis x, y, z, ...podem ser quantificadas, não os parâmetros.

Símbolos não lógicos:

- Símbolos predicativos ou relacionais P, Q, R, ... (possivelmente com índices); Escrevemos P^n para indicar que P é n-ário;
- Símbolos funcionais ou operacionais f, g, h, ...(possivelmente com índices); Escrevemos f^n para indicar que f é n-ário;
- Constantes individuais a, b, c, ... (que não são mais do que símbolos funcionais 0-ários que denotam objectos ou indivíduos fixos).[7]

O conceito de derivabilidade (ou dedutibilidade) deste sistema corresponde à noção de demonstrabilidade da lógica intuicionista.

Os termos e as fórmulas serão definidos como é usual, sendo utilizadas as letras $t, s, ...$ (possivelmente com índices) como metavariáveis para termos, e as letras do alfabeto grego $\varphi, \psi, ...$ como metavariáveis para fórmulas.

1.3.1 Definição

As variáveis, os parâmetros e as constantes são **termos**.

As expressões da forma $ft_1...t_n$, onde $t_1, ..., t_n$ são termos e f é um símbolo funcional n-ário, são termos.

Nada mais é termo.

1.3.2 Definição

Um **termo fechado** é um termo onde não ocorrem variáveis nem parâmetros.

1.3.3 Definição

As regras de formação de **fórmulas** são as seguintes:

1. Se t_1, t_2 são termos, então $t_1 = t_2$ é uma fórmula.
2. Se $t_1, ..., t_n$ são termos e P é um símbolo predicativo n-ário, então $Pt_1...t_n$ é uma fórmula.
3. $\perp$ é uma fórmula.
4. Se φ, ψ são fórmulas, então $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ são fórmulas.
5. Se φ é uma fórmula e x uma variável, então $\forall x\varphi$ e $\exists x\varphi$ são fórmulas.

Nada mais é fórmula .

[7]Convém referir que as letras utilizadas para as variáveis, parâmetros, símbolos de predicados ou funções e constantes são metavariáveis, que representam cada um dos símbolos de $\mathcal{L}$, na metalinguagem relativa à linguagem $\mathcal{L}$.

1.3.4 Definição

As fórmulas dos tipos 1, 2 e 3 dizem-se **atómicas**.

1.3.5 Definição

Uma ocorrência de uma variável x numa fórmula φ diz-se **muda** se essa ocorrência estiver no alcance de um quantificador em x. Caso contrário a ocorrência da variável diz-se **livre**.

Uma variável x diz-se **muda** em φ sse há pelo menos uma ocorrência muda de x na fórmula φ.

1.3.6 Definição

Uma **sentença** é uma fórmula onde não ocorre nenhuma variável livre (parâmetro).

Referimos ainda que a linguagem proposicional intuicionista $\mathcal{L}^0$, além dos conectivos primitivos e dos parênteses, compreende apenas símbolos de relações 0-árias $p, q, r, ...$, possivelmente com índices, designadas por letras proposicionais.

Indicamos de seguida as regras, de eliminação ou introdução, para os conectivos e quantificadores, nas quais se baseia o sistema de dedução natural intuicionista. As regras mais simples são do tipo $\frac{\vdots}{\varphi}$ e significam que das premissas imediatamente acima da linha podemos tirar a conclusão φ. Outras regras são mais complexas e envolvem o uso de hipóteses, que no caso de serem temporárias são *descarregadas* (*eliminadas* ou *canceladas*) no momento da conclusão final, tal como acontece na lógica clássica.

1.3.7 Notação:

A letra caligráfica $\mathcal{D}$ representa uma dada dedução ou derivação.

$$\frac{\begin{array}{c}[\varphi]\\ \mathcal{D}\\ \psi\end{array}}{\varphi \to \psi} \quad \text{ou} \quad \frac{\begin{array}{c}\varphi\,[H]\\ \mathcal{D}\\ \psi\end{array}}{\varphi \to \psi}$$

Esta regra significa que, se existir uma dedução $\mathcal{D}$ que de uma hipótese φ nos permita concluir ψ, então podemos derivar $\varphi \to \psi$.

Por outro lado, o facto de termos colocado φ entre parênteses rectos significa que φ é hipótese relativamente a ψ mas não relativamente a $\varphi \to \psi$, ou seja, esta conclusão final, ao contrário de ψ, já não depende da hipótese φ. Neste caso dizemos que a hipótese temporária φ foi descarregada ou eliminada aquando da aplicação da regra.

Regras de Introdução	Regras de Eliminação
$(\wedge^+)\ \dfrac{\varphi\ \psi}{\varphi\wedge\psi}$	$(\wedge_1^-)\ \dfrac{\varphi\wedge\psi}{\varphi}\ (\wedge_2^-)\ \dfrac{\varphi\wedge\psi}{\psi}$
$(\vee_1^+)\ \dfrac{\varphi}{\varphi\vee\psi}\ (\vee_2^+)\ \dfrac{\psi}{\varphi\vee\psi}$	$(\vee^-)\ \dfrac{\varphi\vee\psi \quad \begin{array}{c}[\varphi]\\ \mathcal{D}_1\\ \sigma\end{array} \quad \begin{array}{c}[\psi]\\ \mathcal{D}_2\\ \sigma\end{array}}{\sigma}$
$(\rightarrow^+)\ \dfrac{\begin{array}{c}[\varphi]\\ \mathcal{D}\\ \psi\end{array}}{\varphi\rightarrow\psi}$	$(\rightarrow^-)\ \dfrac{\varphi\quad\varphi\rightarrow\psi}{\psi}$ (MP)
	$(\perp)\ \dfrac{\perp}{\varphi}$
$(\forall^+)\ \dfrac{\varphi(\alpha)}{\forall x\varphi(x)}$	$(\forall^-)\ \dfrac{\forall x\varphi(x)}{\varphi(t)}$
$(\exists^+)\ \dfrac{\varphi(t)}{\exists x\varphi(x)}$	$(\exists^-)\ \dfrac{\exists x\varphi(x)\quad \begin{array}{c}[\varphi(\alpha_0)]\\ \mathcal{D}\\ \sigma\end{array}}{\sigma}$

1.3.8 Observações:

1. Na regra $(\vee^-)$, a hipótese φ não deve ser utilizada na dedução $\mathcal{D}_2$ e a hipótese ψ não deve ser utilizada na dedução $\mathcal{D}1$.

2. Na regra $(\forall^+)$, $\varphi(\alpha)$ não depende de nenhuma hipótese em que α ocorra e $\varphi(x)$ obtém-se de $\varphi(\alpha)$ substituindo todas as ocorrências de α por x.

3. Na regra $(\forall^-)$, t é um termo fechado e $\varphi(t)$ obtém-se de $\varphi(x)$ substituindo todas as ocorrências de x por t.

4. Na regra $(\exists^+)$, o termo t é fechado e $\varphi(x)$ obtém-se de $\varphi(t)$ substituindo pelo menos uma das ocorrências de t por x.

5. Na regra $(\exists^-)$, α_0 não ocorre na conclusão σ nem nas hipóteses utilizadas na derivação $\mathcal{D}$, excepto na particularização $\varphi(\alpha_0)$ que se obtém substituindo todas as ocorrências livres de x, em $\varphi(x)$, por α_0.

6. A regra $(\rightarrow^-)$ é conhecida classicamente como Modus Ponens (MP).

Vejamos, através de alguns exemplos representativos, que as regras de dedução natural apresentadas estão de acordo com o significado dos conectivos e dos quantificadores, implícito na interpretação BHK.

1.3.9 Exemplos:

1. $(\wedge^+)\ \dfrac{\varphi\ \psi}{\varphi\wedge\psi}$

Sejam A uma demonstração de φ e B uma demonstração de ψ. Então formando o par (A, B) obtemos uma demonstração de $\varphi \wedge \psi$.

2. $(\vee_1^+)$ $\frac{\varphi}{\varphi \vee \psi}$

Seja A uma demonstração de φ. Então $(0, A)$ demonstra $\varphi \vee \psi$.

3. $(\vee^-)$

$$\dfrac{\varphi \vee \psi \quad \begin{array}{c}[\varphi]\\ \mathcal{D}_1\\ \sigma\end{array} \quad \begin{array}{c}[\psi]\\ \mathcal{D}_2\\ \sigma\end{array}}{\sigma}$$

Seja (a, C) uma demonstração de $\varphi \vee \psi$. Estudemos o caso $a = 0$ (o caso $a = 1$ seria análogo).

Se $a = 0$, então (a, C) demonstra φ. Por outro lado, a dedução $\mathcal{D}_1$ fornece-nos um processo (construtivo) de encontrar uma demonstração de σ partindo da demonstração de φ. Sendo assim temos uma construção que transforma uma demonstração de φ numa demonstração de σ, ou seja, temos uma demonstração, digamos B, de $\varphi \to \sigma$. Pela interpretação BHK, $B(a, C)$ é uma demonstração de σ.

4. $(\perp)$ $\frac{\perp}{\varphi}$

A justificação desta regra em termos de construções não é óbvia, uma vez que concluimos φ sem apresentar uma demonstração sua. No entanto, reparemos que a regra está de acordo com a concepção intuicionista, na medida em que é justificada pela interpretação BHK. Segundo esta interpretação nenhuma construção demonstra $\perp$. Para justificar a regra, dada uma demonstração de $\perp$ deveríamos encontrar uma construção que a transformasse numa demonstração de φ. Ora, como não existe uma demonstração de $\perp$, não é necessário procurar a referida construção. Portanto, a regra é válida e podemos concluir φ sem que seja apresentada uma sua construção demonstrativa. É importante referir que esta regra não é uma demonstração de φ mas sim uma demonstração de que podemos concluir φ *dentro* de um raciocínio em que a determinada altura obtemos uma demonstração de $\perp$ (que não pode existir).

1.3.10 Definição

ψ é **dedutível** ou **derivável** de $\varphi_1, ..., \varphi_n$, e escrevemos $\varphi_1, ..., \varphi_n \vdash \psi$, se existir uma sequência finita de fórmulas $\varphi_1, ..., \varphi_n, \varphi_{n+1}, ..., \psi$ (**derivação** ou **dedução** formal) em que cada uma delas é uma das n primeiras (**hipóteses** da dedução), ou uma hipótese descarregada, ou é inferida de uma (ou mais) das fórmulas que a precedem, por uma das regras de inferência do sistema. A conclusão final ψ designa-se por **tese**.

1.3.11 Definição

Dado um conjunto (finito ou infinito) Γ de fórmulas de $\mathcal{L}$ e uma fórmula ψ, dizemos que ψ é **dedutível** (ou **derivável** ou **teorema**) de Γ, e escrevemos $\Gamma \vdash \psi$, se existirem $\varphi_1, ..., \varphi_n$ em Γ, com $n \geq 0$, tais que $\varphi_1, ..., \varphi_n \vdash \psi$.

Consideramos a possibilidade de se ter $n = 0$ para admitirmos a existência de deduções sem hipóteses.

1.3.12 Notação:

1. Utilizamos Γ,φ para abreviar $\Gamma \cup \varphi$.

2. Sendo este sistema de dedução natural diferente do clássico, também o conceito de derivabilidade que lhe está associado é diferente e, portanto, deveríamos alterar a notação usual $\vdash$ para, por exemplo, $\vdash_i$.

No entanto, quando o contexto não deixar dúvidas acerca da lógica subjacente, continuaremos a utilizar o símbolo $\vdash$ e a falar de validade de uma dedução, em vez de validade intuicionista de uma dedução.

1.3.13 Observação

Daqui em diante, para apresentar uma derivação de determinada tese, eventualmente com hipóteses, optamos por uma disposição vertical da dedução em detrimento de uma disposição em árvore, por nos parecer que se torna mais fácil a sua leitura e verificação.

Uma vez que o sistema de dedução natural clássico difere do intuicionista apenas na aceitação da regra $(\neg\neg^-)$, uma dedução clássica que não utilize esta regra, nem nenhuma que lhe seja equivalente, nomeadamente a redução ao absurdo (RA), será intuicionisticamente válida desde que as permissas o sejam (demonstração por indução no comprimento das deduções).[8]

Apresentamos uma lista de resultados, com a respectiva dedução, que poderá eventualmente ser útil.

1.3.14 Lema

1. $\vdash \bot \leftrightarrow (\varphi \wedge \neg\varphi)$

2. $\varphi \vdash \neg\neg\varphi$

3. $\vdash \neg\varphi \leftrightarrow \neg\neg\neg\varphi$

4. $\varphi \vdash (\psi \rightarrow \varphi)$

---

[8]Uma dedução que inclua a lei do terceiro excluído não é intuicionisticamente válida. A demonstração clássica da lei do terceiro excluído inclui uma aplicação da regra $(\neg\neg^-)$.

5. $\varphi \vdash (\neg\varphi \to \psi)$

6. $\varphi \to \psi, \neg\psi \vdash \neg\varphi$

7. $\vdash \neg\exists x\varphi(x) \leftrightarrow \forall x\neg\varphi(x)$

8. $\vdash \exists x\neg\varphi(x) \to \neg\forall x\varphi(x)$

9. $\exists x(\varphi \to \psi(x)) \vdash \varphi \to \exists x\psi(x)$

Demonstração

1. $\vdash \bot \leftrightarrow (\varphi \wedge \neg\varphi)$

$\vdash \bot \to (\varphi \wedge \neg\varphi)$ resulta de uma aplicação directa da regra $(\bot)$ seguida de uma aplicação da regra $(\to^+)$.

Resta verificar que $\vdash (\varphi \wedge \neg\varphi) \to \bot$.

1	$\varphi \wedge \neg\varphi$	$[H_1]$
2	φ	$1, (\wedge_1^-)$
3	$\varphi \to \bot$	$1(\wedge_2^-)$
4	$\bot$	$2, 3(\to^-)$
5	$(\varphi \wedge \neg\varphi) \to \bot$	$1-4(\to^+)$

Nota: A regra $(\bot^+)$ $\frac{\varphi \;\; \neg\varphi}{\bot}$ é conhecida como regra de introdução do absurdo.

2. $\varphi \vdash \neg\neg\varphi$

1	φ	H_1
2	$\neg\varphi$	$[H_2]$
3	$\varphi \wedge \neg\varphi$	$1, 2(\wedge^+)$
4	$\varphi \wedge \neg\varphi \to \bot$	(lei 1)
5	$\bot$	$3, 4(\to^-)$
6	$\neg\neg\varphi$	$2-5(\to^+)$

Nota: Sempre que, como se verifica neste caso, temos uma dedução com hipóteses, podemos, através de uma ou várias aplicações da regra $(\to^+)$, descarregá-las e obter uma derivação sem hipóteses. No caso anterior bastaria acrescentar a linha 7 $\varphi \to \neg\neg\varphi$ $1-6(\to^+)$ e eliminar a hipótese H_1 para obter uma demonstração de $\vdash \varphi \to \neg\neg\varphi$.

3. $\vdash \neg\varphi \leftrightarrow \neg\neg\neg\varphi$

$\neg\varphi \to \neg\neg\neg\varphi$ é um caso particular da lei que demonstrámos no ponto anterior (ver nota).

Resta verificar que $\vdash \neg\neg\neg\varphi \to \neg\varphi$.

1	$\neg\neg\neg\varphi$	$[H_1]$
2	$\varphi\to\neg\neg\varphi$	(lei 2)
3	φ	$[H_2]$
4	$\neg\neg\varphi$	$2,3(\to^-)$
5	$\neg\neg\neg\varphi\wedge\neg\neg\varphi$	$1,4(\wedge^+)$
6	$\neg\neg\neg\varphi\wedge\neg\neg\varphi\to\bot$	(lei 1)
7	$\bot$	$5,6(\to^-)$
8	$\neg\varphi$	$3-7(\to^+)$
9	$\neg\neg\neg\varphi\to\neg\varphi$	$1-6(\to^+)$

4. $\varphi\vdash(\psi\to\varphi)$

1	φ	H_1
2	ψ	$[H_2]$
3	$\varphi\wedge\psi$	$1,2(\wedge^+)$
4	φ	$3(\wedge_1^-)$
5	$\psi\to\varphi$	$2-4(\to^+)$

5. $\varphi\vdash(\neg\varphi\to\psi)$

1	φ	H_1
2	$\neg\varphi$	$[H_2]$
3	$\varphi\wedge\neg\varphi$	$1,2(\wedge^+)$
4	ψ	$3(\bot)$
5	$\neg\varphi\to\psi$	$2-4(\to^+)$

6. $\varphi\to\psi,\neg\psi\vdash\neg\varphi$

1	$\varphi\to\psi$	H_1
2	$\neg\psi$	H_2
3	φ	$[H_3]$
4	ψ	$1,3(\to^-)$
5	$\bot$	$2,4(\bot^+)$
6	$\neg\varphi$	$3-5(\to^+)$

Nota: A regra (MT) $\frac{\varphi\to\psi\quad\neg\psi}{\neg\varphi}$ é conhecida como Modus Tollens.

7. $\vdash\neg\exists x\varphi(x)\leftrightarrow\forall x\neg\varphi(x)$

$\vdash\neg\exists x\varphi(x)\to\forall x\neg\varphi(x)$

1	$\neg\exists x\varphi(x)$	$[H_1]$
2	$\varphi(\alpha)$	$[H_2]$
3	$\exists x\varphi(x)$	$2,(\exists^+)$
4	$\neg\exists x\varphi(x)\wedge\exists x\varphi(x)$	$1,3(\wedge^+)$
5	$\neg\exists x\varphi(x)\wedge\exists x\varphi(x)\to\bot$	(lei 1)
6	$\bot$	$4,5(\to^-)$
7	$\neg\varphi(\alpha)$	$2-4(\to^+)$
8	$\forall x\neg\varphi(x)$	$5(\forall^+)$
9	$\neg\exists x\varphi(x)\to\forall x\neg\varphi(x)$	$1-6(\to^+)$

$\vdash \forall x \neg\varphi(x) \to \neg\exists x \varphi(x)$

1	$\forall x \neg\varphi(x)$	$[H_1]$
2	$\exists x \varphi(x)$	$[H_2]$
3	$\varphi(\alpha_0)$	$[H_3]$
4	$\neg\varphi(\alpha_0)$	$1(\forall^-)$
5	$\varphi(\alpha_0) \wedge \neg\varphi(\alpha_0)$	$3,4(\wedge^+)$
6	$\varphi(\alpha_0) \wedge \neg\varphi(\alpha_0) \to \bot$	(lei 1)
7	$\bot$	$5,6(\to^-)$
8	$\bot$	$2,3-7(\exists^-)$
9	$\neg\exists x \varphi(x)$	$2-8(\to^+)$
10	$\forall x \neg\varphi(x) \to \neg\exists x \varphi(x)$	$1-9(\to^+)$

8. $\vdash \exists x \neg\varphi(x) \to \neg\forall x \varphi(x)$

1	$\exists x \neg\varphi(x)$	$[H_1]$
2	$\forall x \varphi(x)$	$[H_2]$
3	$\neg\varphi(\alpha_0)$	$[H_3]$
4	$\varphi(\alpha_0)$	$2(\forall^-)$
5	$\neg\varphi(\alpha_0) \wedge \varphi(\alpha_0)$	$3,4(\wedge^+)$
6	$\neg\varphi(\alpha_0) \wedge \varphi(\alpha_0) \to \bot$	(lei)
7	$\bot$	$5,6(\to^-)$
8	$\bot$	$1,3-7(\exists^-)$
9	$\neg\forall x \varphi(x)$	$2-8(\to^+)$
10	$\exists x \neg\varphi(x) \to \neg\forall x \varphi(x)$	$1-9(\to^+)$

9. $\exists x(\varphi \to \psi(x)) \vdash \varphi \to \exists x \psi(x)$

1	$\exists x(\varphi \to \psi(x))$	H_1
2	φ	$[H_2]$
3	$\varphi \to \psi(\alpha_0)$	$[H_3]$
4	$\psi(\alpha_0)$	$2,3(\to^-)$
5	$\exists x \psi(x)$	$4(\exists^+)$
6	$\exists x \psi(x)$	$1,3-5(\exists^-)$
7	$\varphi \to \exists x \psi(x)$	$2-6(\to^+)$

■

1.3.15 Observação:

Recordemos as regras clássicas de introdução da negação $(\neg^+)$ e de redução ao absurdo (RA) ou $(\neg^-)$:

$$(\neg^+)\ \frac{\begin{matrix}\varphi\ [H] \\ \mathcal{D} \\ \bot\end{matrix}}{\neg\varphi} \qquad (\text{RA})\ \frac{\begin{matrix}\neg\varphi\ [H] \\ \mathcal{D} \\ \bot\end{matrix}}{\varphi}$$

A regra clássica $(\neg^{+})$ é intuicionisticamente válida pois resulta de uma aplicação da regra $(\rightarrow^{+})$. Contudo, a regra de redução ao absurdo não é admitida pelos intuicionistas pois a sua dedução, além da regra $(\rightarrow^{+})$, utiliza também a eliminação da dupla negação $(\neg\neg^{-})$.

Capítulo 2

Semântica de Kripke

2.1 Semântica para a lógica intuicionista

Referimos anteriormente que na lógica intuicionista uma sentença é válida se temos uma demonstração construtiva para ela e é falsa se a hipótese de que existe uma prova da sentença nos conduz a uma contradição. Nesse sentido, e de acordo com Brouwer, o significado de "validade intuicionista" é "demonstrabilidade construtiva".

Heyting e Kolmogorov explicitaram essa definição de validade intuicionista através da semântica BHK que introduzimos na secção *1.2*. No entanto, a semântica BHK é uma interpretação informal que utiliza as noções de construção e prova (ou demonstração) como primitivas, não sendo portanto definidas.

Actualmente são conhecidas diferentes semânticas da lógica intuicionista[9], que se revelam adequadas na medida em que permitem estabelecer resultados de validade e completude (**Artemov, 1998** e **1999**). A formalização da semântica BHK seria essencial para estabelecer a completude da lógica intuicionista com respeito a esta interpretação.

Na secção que se segue vamos introduzir uma interpretação semântica diferente, que não está directamente relacionada com a interpretação BHK. Esta semântica deve-se a Kripke e, por se revelar adequada à lógica intuicionista[10], vai ser muito útil na demonstração da não derivabilidade intuicionista de certos resultados.

Recordemos que, um modelo para o cálculo proposicional clássico, consiste numa valuação das proposições com valores 0 ou 1. Obtemos dessa forma o estado do nosso conhecimento: um conjunto conhecido de sentenças verdadeiras e um conjunto conhecido de sentenças falsas (sendo que cada uma delas ou é falsa ou é verdadeira).

Na semântica para a lógica intuicionista proposta por Kripke, o nosso conhecimento acerca de cada sentença e da sua validade pode melhorar. Admite-se a possibilidade da validade de uma sentença ser indeterminada num determinado estádio do nosso conhecimento e vir a ser estabelecida como certa num estádio de conhecimento posterior. O significado de *verdade* é portanto "já estabelecido como verdade" (verdade essa que será preservada com o avanço do conhecimento). O significado de *falso* é "ainda não estabelecido como verdadeiro".

[9]Segundo Troelstra (**Barwise, 1993**) as interpretações para a lógica intuicionista incluem-se num de dois tipos: modificações do esquema BHK (ex: realizabilidade) ou semânticas do tipo "truth-value" (ex: modelos de Beth, modelos de Kripke...).
[10]De referir que inicialmente a semântica de Kripke foi formulada para lógicas modais.

2.2 Modelos de Kripke

Parece-nos consensual a ideia de que um matemático pode, ao longo do tempo, aumentar o universo de objectos que conhece e o número de leis que estabeleceu até ao momento. Consideremos então que temos diferentes estádios de conhecimento, que num determinado estádio, k, temos um conjunto de objectos que construímos e um conjunto de sentenças que estabelecemos, e que num próximo estádio das nossas actividades mentais temos várias escolhas possíveis.

Um modelo de Kripke para a lógica intuicionista satisfaz o *Princípio da preservação do conhecimento*: Se uma sentença é válida num determinado estádio, então também é válida em todos os estádios acessíveis a partir dele. Sendo assim, os estádios de actividade mental que referimos devem ser parcialmente ordenados.

Antes de procedermos à formalização da semântica de Kripke procuremos compreender a interpretação dos conectivos que lhe é subjacente.

Como seria de esperar, em nenhum estádio podemos estabelecer a validade de $\perp$.

Se num determinado estádio tivermos estabelecido a validade de φ e a validade de ψ então podemos estabelecer a validade de $\varphi \wedge \psi$. Analogamente, se num determinado estádio tivermos estabelecido a validade de φ ou a validade de ψ então podemos estabelecer a validade de $\varphi \vee \psi$. O conectivo $\rightarrow$ tem uma interpretação diferente já que a validade de $\varphi \rightarrow \psi$ pode ser estabelecida num estádio sem que φ ou ψ tenham sido estabelecidas. Podemos estabelecer a validade de $\varphi \rightarrow \psi$, no estádio k, se soubermos que sempre que num estádio futuro (incluindo o k) se estabelecer φ então também se estabelece ψ.

A validade de $\forall x \varphi(x)$ é estabelecida, no estádio k, se em qualquer estádio futuro (incluindo o k) para todos os objectos d existentes nesse estádio, $\varphi(d)$ seja estabelecido. Contrariamente ao que acontece com o quantificador universal, em relação ao quantificador existencial não é necessário ter em conta o futuro, uma vez que o matemático construtivista apenas afirma a existência de um objecto com determinadas propriedades quando já o construiu. Sendo assim, a validade de $\exists x \varphi(x)$ é estabelecida, no estádio k, se já construímos um objecto a tal que no estádio k se estabeleça $\varphi(a)$.

Apresentamos em seguida a definição de *Modelo de Kripke* para a lógica de predicados (introduzida inicialmente em 1965) com uma linguagem que contém apenas símbolos predicativos ou relacionais.

2.2.1 Definição

Um **modelo de Kripke** para a lógica de predicados intuicionista é um quádruplo $\mathfrak{K} \equiv (K, \leq, D, \vDash)$, com K não vazio, tal que:

(i) $(K, \leq)$ é um conjunto parcialmente ordenado.

(ii) D é uma função (*função domínio*) que a cada elemento, k, de K, faz corresponder um conjunto não vazio $D(k)$, tal que: Quaisquer que sejam k e k' pertencentes a K, se $k \leq k'$ então $D(k) \subseteq D(k')$. Ou seja, D é monótona crescente (em sentido lato). Ao conjunto $D(k)$ chamamos domínio de $\mathfrak{K}$ no estádio k.

(iii) $\vDash$ é uma relação binária em $K \times \mathcal{P}$, sendo $\mathcal{P}$ o conjunto das fórmulas atómicas, tal que: Se $k \vDash P$ e $k' \geq k$ então $k' \vDash P$.

Nota:

- K pode ser um conjunto finito pois num determinado estádio podemos optar por parar em vez de prosseguir para um dos estádios acessíveis a partir desse.
- Os elementos de K designam-se por *nós* do modelo $\mathfrak{K}$.
- $k \vDash P$ lê-se "k força P" ou "P é verdadeira no nó k".
- No caso da linguagem que estamos a utilizar, P pode ser uma letra proposicional (símbolo de relação 0-ária) ou um símbolo predicativo n-ário $Pt_1...t_n$ em que os termos apenas podem ser as constantes $a, b, c, d, ...$, (elementos dos $D(k)$ que identificamos com os símbolos que os representam) possivelmente com índices. Portanto, se $k \vDash P^n(d_1, d_2, ..., d_n)$ então $d_i \in D(k)$ para $1 \leq i \leq n$.

2.2.2 Notação:

Utilizamos $k \nvDash P$ com o significado "k não força P" ou "P não é verdadeira no nó k".

A relação $\vDash$ extende-se às fórmulas compostas da seguinte forma:

K₁. Não existe nenhum elemento $k \in K$ tal que $k \vDash \bot$.

K₂. $k \vDash \varphi \wedge \psi := k \vDash \varphi$ e $k \vDash \psi$

K₃. $k \vDash \varphi \vee \psi := k \vDash \varphi$ ou $k \vDash \psi$

K₄. $k \vDash \varphi \rightarrow \psi :=$ Para todo o k' pertencente a K, se $k' \geq k$ e $k' \vDash \varphi$ então $k' \vDash \psi$.

K₅. $k \vDash \forall x \varphi(x) := \boldsymbol{\forall} k' \geq k \boldsymbol{\forall} d \in D(k')\big(k' \vDash \varphi(d)\big)$[11]

K₆. $k \vDash \exists x \varphi(x) := \boldsymbol{\exists} d \in D(k)\big(k \vDash \varphi(d)\big)$

[11] É importante referir que por vezes, como neste caso, cometemos abusos ao utilizar quantificadores na metalinguagem por uma questão de brevidade e simplificação de escrita. Uma vez que explicitámos a interpretação dos conectivos nesta semântica, julgamos não ser difícil, em cada caso, distinguir os quantificadores na linguagem objecto ou na metalinguagem. No entanto, apresentamos a negrito os símbolos $\boldsymbol{\forall}$ e $\boldsymbol{\exists}$ na metalinguagem, evitando uma possível confusão.

Vejamos algumas consequências desta definição.

2.2.3 Lema

$k \vDash \neg\varphi$ sse para todo $k' \geq k$, $k' \nvDash \varphi$.

Demonstração

$k \vDash \neg\varphi$ sse $k \vDash \varphi \rightarrow \bot$ sse $\forall k' \geq k$, $(k' \vDash \varphi \Rightarrow k' \vDash \bot)$ sse $\forall k' \geq k$, $k' \nvDash \varphi$.[12]

2.2.4 Lema

$k \vDash \neg\neg\varphi$ sse $\forall k' \geq k\ \exists k'' \geq k'(k'' \vDash \varphi)$

Demonstração

Suponhamos que estabelecemos $\forall k' \geq k\ \ \exists k'' \geq k'(k'' \vDash \varphi)$. Aplicando o Lema 1, $\forall k' \geq k$ não se verifica $(k' \vDash \neg\varphi)$, ou seja $\forall k' \geq k$, $k' \nvDash \neg\varphi$. Portanto $k \vDash \neg\neg\varphi$, novamente pelo Lema 1.

Na demonstração da implicação recíproca utilizamos um raciocínio pouco intuicionista ao nível da metalinguagem.

Suponhamos que $k \vDash \neg\neg\varphi$. Então pelo Lema 1, para todo $k' \geq k$, $k' \nvDash \neg\varphi$. Consequentemente, por aplicação do mesmo Lema, para todo $k' \geq k$, *existe* pelo menos um $k'' \geq k'$ tal que $k'' \vDash \varphi$. ■

2.2.5 Lema (Monotonia de $\vDash$)

$\forall k, k' \in K$(se $k' \geq k$ e $k \vDash \varphi$ então $k' \vDash \varphi$), qualquer que seja a fórmula φ.

Demonstração

Demonstremos o lema por indução na complexidade das fórmulas.

I_1.Pela definição 1 desta secção, as fórmulas atómicas verificam esta propriedade.

I_2.Suponhamos que φ verifica a propriedade. Provemos que $\neg\varphi$ também a verifica.

Fixemos $k \in K$ tal que $k \vDash \neg\varphi$. Então $\forall k' \geq k$ (k' não força φ). Logo, $\forall k'' \geq k'$ (k'' não força φ) pois $k'' \geq k$. Sendo assim, $k' \vDash \neg\varphi$.

I_3.Suponhamos que φ e ψ verificam a propriedade.

Fixemos $k \in K$ tal que $k \vDash \varphi \wedge \psi$. Então por $\mathbf{K_2}$ $k \vDash \varphi$ e $k \vDash \psi$. Segue-se, por hipótese de indução, que $\forall k' \geq k$ $(k' \vDash \varphi)$ e $\forall k' \geq k$ $(k' \vDash \psi)$, ou seja $\forall k' \geq k$ $(k' \vDash \varphi$ e $k' \vDash \psi)$. Novamente por $\mathbf{K_2}$ $k' \vDash \varphi \wedge \psi$.

---

[12]Utilizamos o símbolo $\Rightarrow$ como abreviatura da expressão "Se ... então ...".

Fixemos $k \in K$ tal que $k \vDash \varphi \vee \psi$. Então por $\mathbf{K_3}$ $k \vDash \varphi$ ou $k \vDash \psi$. Por hipótese de indução, $\forall k' \geq k$ $(k' \vDash \varphi)$ ou $\forall k' \geq k$ $(k' \vDash \psi)$, logo $\forall k' \geq k$ $(k' \vDash \varphi$ ou $k' \vDash \psi)$. Novamente por $\mathbf{K_3}$ $k' \vDash \varphi \vee \psi$.

Fixemos $k \in K$ tal que $k \vDash \varphi \rightarrow \psi$. Então, por $\mathbf{K_4}$, $\forall k' \geq k$ $(k' \vDash \varphi \Rightarrow k' \vDash \psi)$. Tomemos $k'' \geq k'$. Suponhamos que $k'' \vDash \varphi$. Então $k'' \vDash \psi$, pois $k'' \geq k$. Como k'' é um elemento arbitrário de K maior do que k' podemos estabelecer que $\forall k'' \geq k'$ $(k'' \vDash \varphi \Rightarrow k'' \vDash \psi)$, ou seja, $k' \vDash \varphi \rightarrow \psi$, novamente por $\mathbf{K_4}$.

I_4.Fixemos $k \in K$ tal que $k \vDash \forall x \varphi(x)$. Então, por $\mathbf{K_5}$, para todo $k' \geq k$, $\forall d \in D(k')(k' \vDash \varphi(d))$. Em particular, para todo $k'' \geq k'$, $k'' \vDash \varphi(b)$, qualquer que seja $b \in D(k'')$, já que $k'' \geq k$. Logo $k' \vDash \forall x \varphi(x)$, novamente por $\mathbf{K_5}$.

Suponhamos que φ verifica a propriedade. Fixemos $k \in K$ tal que $k \vDash \exists x \varphi(x)$. Seja $k' \geq k$. Por $\mathbf{K_6}$, $\exists d \in D(k)(k \vDash \varphi(d))$. Como $k' \geq k$, $D(k) \subset D(k')$ e portanto $d \in D(k')$. Ora φ verifica a propriedade, logo $k' \vDash \varphi(d)$ e consequentemente $\exists d \in D(k')(k' \vDash \varphi(d))$, ou seja $k' \vDash \exists x \varphi(x)$. ■

Para especificarmos um modelo de Kripke vamos utilizar um diagrama em árvore que simboliza a estrutura parcialmente ordenada, escrevendo ao lado de cada nó,k, do modelo, o conjunto $D(k)$ e as fórmulas atómicas verdadeiras nesse nó.

2.2.6 Observação

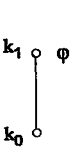
Veremos adiante que uma fórmula é válida num modelo sse for válida no seu primeiro nó. Pelo Teorema da Validade, se uma fórmula não for válida num modelo então não é dedutível na lógica intuicionista.

Em cada um dos exemplos que se seguem apresentamos um modelo em que o primeiro nó não força uma determinada lei clássica. Sendo assim, com um contra-exemplo simples mas adequado, provamos que fórmulas dedutíveis na lógica clássica não são dedutíveis para os intuicionistas.

2.2.7 Exemplos:

Comecemos com alguns modelos na lógica proposicional. Nesse caso, ao lado de cada nó, apresentamos apenas as fórmulas atómicas verdadeiras nesse estádio.

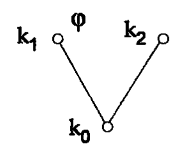
1. $k_0 \nvDash \neg\neg\varphi \rightarrow \varphi$ e $k_0 \nvDash \varphi \vee \neg\varphi$



Neste modelo de Kripke, no nó k_0 ainda não é conhecida nenhuma fórmula atómica mas no nó k_1 é conhecida φ. Logo $k_0 \not\models \varphi$ mas $k_1 \models \varphi$. Pelo Lema 2, $k_0 \models \neg\neg\varphi$. Sendo assim podemos concluir que $k_0 \not\models \neg\neg\varphi \to \varphi$.

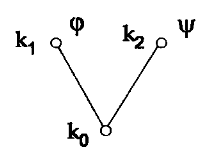
Por outro lado, pelo lema 1, $k_0 \not\models \neg\varphi$ visto que $k_1 \models \varphi$. Logo, como φ também não é verdadeira no nó k_0, podemos concluir que $k_0 \not\models \varphi \vee \neg\varphi$.

2. $k_0 \not\models (\neg\varphi \vee \neg\neg\varphi)$



Neste modelo $k_1 \models \varphi$. Pelo Lema 1 concluímos que $k_2 \models \neg\varphi$. Então, $k_0 \not\models \neg\varphi$ porque $k_1 \models \varphi$ e $k_0 \not\models \neg\neg\varphi$ porque $k_2 \models \neg\varphi$. Logo $k_0 \not\models (\neg\varphi \vee \neg\neg\varphi)$.

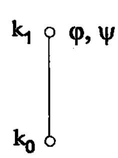
3. $k_0 \not\models \neg(\varphi \wedge \psi) \to (\neg\varphi \vee \neg\psi)$



Nenhum nó deste modelo força $\varphi \wedge \psi$. Logo $k_0 \models \neg(\varphi \wedge \psi)$. No entanto, $k_0 \not\models (\neg\varphi \vee \neg\psi)$, visto que $k_0 \not\models \neg\varphi$ e $k_0 \not\models \neg\psi$.

Portanto, $k_0 \not\models \neg(\varphi \wedge \psi) \to (\neg\varphi \vee \neg\psi)$.

4. $k_0 \not\models (\psi \to \varphi) \to (\neg\psi \vee \varphi)$

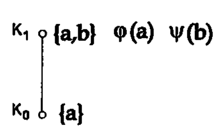


Neste modelo $k_0 \models (\psi \to \varphi)$. No entanto, $k_0 \not\models \neg\psi$ pois $k_1 \models \psi$ e $k_0 \not\models \varphi$, logo $k_0 \not\models \neg\psi \vee \varphi$.

Portanto, $k_0 \not\models (\psi \to \varphi) \to (\neg\psi \vee \varphi)$.

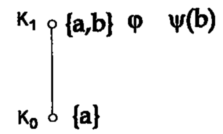
Vejamos agora alguns modelos na lógica de predicados.

5. $k_0 \not\models \neg\forall x \varphi(x) \to \exists x \neg\varphi(x)$



Observemos que $k_0 \nVdash \forall x\varphi(x)$, pois $k_0 \nVdash \varphi(a)$, e que $k_1 \nVdash \forall x\varphi(x)$ visto que $k_1 \nVdash \varphi(b)$. Então $k_0 \Vdash \neg\forall x\varphi(x)$. No entanto, $k_0 \nVdash \exists x\neg\varphi(x)$ já que $D(k_0) = \{a\}$ e $k_0 \nVdash \neg\varphi(a)$ porque $k_1 \Vdash \varphi(a)$. Logo concluímos que $k_0 \nVdash \neg\forall x\varphi(x) \to \exists x\neg\varphi(x)$.[13]

6. $k_0 \nVdash (\varphi \to \exists x\psi(x)) \to \exists x(\varphi \to \psi(x))$



$k_0 \Vdash \varphi \to \exists x\psi(x)$, visto que o único nó que força φ é k_1 e $k_1 \Vdash \exists x\psi(x)$ pois $k_1 \Vdash \psi(b)$. Reparemos, no entanto, que $k_0 \nVdash \varphi \to \psi(a)$, já que $k_1 \Vdash \varphi$ mas $k_1 \nVdash \psi$(a). Logo, como $D(k_0) = \{a\}$ concluímos que $k_0 \nVdash \exists x(\varphi \to \psi(x))$.

Portanto, $k_0 \nVdash (\varphi \to \exists x\psi(x)) \to \exists x(\varphi \to \psi(x))$. ■

2.3 Validade e Completude

2.3.1 Definição

(i) Uma fórmula F é válida no nó k de um modelo de Kripke $\mathfrak{K}$ sse $k \Vdash F$.

(ii) F é válida no modelo $\mathfrak{K} \equiv (K, \leq, D, \Vdash)$ sse $k \Vdash F$, para todo o $k \in K$.

2.3.2 Observação

Notemos que, pelo Lema 2.2.5 (Monotonia de $\Vdash$), considerando k_0 o mínimo de K, F é válida no modelo $\mathfrak{K}$ sse $k_0 \Vdash F$.

2.3.3 Notação

Se F é válida no modelo $\mathfrak{K}$ escrevemos $\mathfrak{K} \Vdash F$.

Se Γ é um conjunto de fórmulas, ultilizamos a notação $\mathfrak{K} \Vdash \Gamma$ para abreviar $\mathfrak{K} \Vdash A$ qualquer que seja a fórmula $A \in \Gamma$.

2.3.4 Definição

(i) Seja Γ um conjunto de fórmulas. F é uma Kripke-consequência de Γ (escrevemos $\Gamma \Vdash F$) sse em qualquer modelo $\mathfrak{K}$ onde se verifique $\mathfrak{K} \Vdash \Gamma$, também se verifique $\mathfrak{K} \Vdash F$.

(ii) F é Kripke-válida (K-válida) sse $\emptyset \Vdash F$, ou seja, se e só se F é válida em qualquer modelo.

Se F é K-válida escrevemos simplesmente $\Vdash F$.

[13]É de referir que $\psi(b)$ não foi utilizado para estabelecer o que pretendíamos e, portanto, poderíamos omitir $\psi(b)$ neste exemplo. No entanto, é importante notar que podemos ter conhecimento de uma fórmula num estádio e não utilizarmos esse conhecimento num determinado raciocínio.

2.3.5 Observação

Em cada um dos exemplos que vimos anteriormente encontramos um modelo cujo primeiro nó não força uma lei clássica. Não sendo válida no primeiro nó, a lei não é válida no modelo apresentado. Foram então apresentados contra-exemplos que mostram que as referidas leis clássicas não são K-válidas.

2.3.6 Teorema (Validade ou Adequação)

Seja Γ um conjunto de fórmulas. Então $\Gamma \vdash F \Rightarrow \Gamma \vDash F$. Ou seja, se F é derivável de um conjunto de fórmulas Γ, então F é uma Kripke-consequência de Γ.

Em particular, se $\vdash F$ então $\vDash F$. Ou seja, se F é uma lei dedutível na lógica intuicionista, então F é K-válida.

Demonstração

Demonstremos o teorema por indução na derivação, $\mathcal{D}$, de F com hipóteses em Γ.

Fixemos um modelo de Kripke $\mathfrak{K} = (K, \leq, D, \vDash)$.

Vamos utilizar a notação $k \vDash \Gamma$ para abreviar $k \vDash A$, para toda a fórmula $A \in \Gamma$.

I_1. Se a derivação $\mathcal{D}$ consistir apenas na fórmula F, então $F \in \Gamma$. Nesse caso é imediato que $k \vDash \Gamma \Rightarrow k \vDash F$, para todo o $k \in K$.

I_2. Suponhamos agora que $\mathcal{D}$ termina com a aplicação de uma determinada regra de derivação. Estudemos os diferentes casos possíveis. Fixemos $k \in K$, tal que $k \vDash \Gamma$.

$(\wedge^+)$ Se $\mathcal{D}$ termina com a aplicação desta regra então F é do tipo $\varphi_1 \wedge \varphi_2$.

Por hipótese de indução, $k \vDash \varphi_1$ e $k \vDash \varphi_2$. Logo $k \vDash \varphi_1 \wedge \varphi_2$. Concluímos assim que $\Gamma \vDash F$.

$(\wedge^-)$ Suponhamos que F foi obtida de $\varphi_1 \wedge \varphi_2$ por aplicação de uma das regras $(\wedge^-)$. Por hipótese de indução, $k \vDash \varphi_1 \wedge \varphi_2$. Logo $k \vDash \varphi_1$ e $k \vDash \varphi_2$. Portanto $k \vDash F$.

$(\vee_1^+)$ Neste caso F é do tipo $\varphi_1 \vee \varphi_2$, obtida de φ_1 por aplicação de $(\vee_1^+)$. Por hipótese de indução, $k \vDash \varphi_1$. Logo podemos concluir que $k \vDash \varphi_1$ ou $k \vDash \varphi_2$. Portanto $k \vDash F$.

$(\vee^-)$ Suponhamos que F foi obtida de $\varphi_1 \vee \varphi_2$ por aplicação desta regra. Por hipótese de indução:

1. $k \vDash \Gamma \Rightarrow k \vDash (\varphi_1 \vee \varphi_2)$
2. $k \vDash \Gamma, \varphi_1 \Rightarrow k \vDash F$
3. $k \vDash \Gamma, \varphi_2 \Rightarrow k \vDash F$

Como fixámos k tal que $k \models \Gamma$, concluímos que $k \models (\varphi_1 \vee \varphi_2)$, por 1. Logo, $k \models \varphi_1$ ou $k \models \varphi_2$.

No primeiro caso, temos que $k \models \varphi_1$ e $k \models \Gamma$, ou seja, $k \models \varphi_1 \cup \Gamma$. Então, $k \models F$, por 2. No segundo caso, temos que $k \models \varphi_2$ e $k \models \Gamma$, ou seja, $k \models \varphi_2 \cup \Gamma$. Então, $k \models F$, por 3. Em qualquer dos casos, $k \models F$. Portanto, $\Gamma \models F$.

($\to^+$) Neste caso F é do tipo $\varphi_1 \to \varphi_2$. Por hipótese de indução, $k \models \Gamma, \varphi_1 \Rightarrow k \models \varphi_2$, qualquer que seja $k \in K$.

Queremos provar que $k \models \varphi_1 \to \varphi_2$. Tomemos $k' \geq k$ tal que $k' \models \varphi_1$.

Pela propriedade da monotonia, se $k \models \Gamma$ então $k' \models \Gamma$. Temos então que $k' \models \Gamma$ e $k' \models \varphi_1$, ou seja, $k' \models \Gamma \cup \varphi_1$. Logo, pela hipótese de indução, $k' \models \varphi_2$. Fica assim provado que $k \models \varphi_1 \to \varphi_2$, ou seja, $k \models F$.

($\to^-$) Suponhamos que F foi obtida de $\varphi_1 \to \varphi_2$ e de φ_1, por aplicação desta regra. Por hipótese de indução, $k \models (\varphi_1 \to \varphi_2)$ e $k \models \varphi_1$.

Sendo assim, para todo $k' \geq k$, se $k' \models \varphi_1$ então $k' \models \varphi_2$. Em particular, também para k, $k \models \varphi_1 \Rightarrow k \models \varphi_2$. Logo, $k \models \varphi_2$, ou seja, $k \models F$.

($\perp$) Suponhamos que a dedução $\mathcal{D}$, com hipóteses em Γ, termina com a aplicação desta regra. Por hipótese de indução, para todo o $k \in K$, $(k \models \Gamma \Rightarrow k \models \perp)$. Como nenhum nó pode forçar $\perp$, nenhum nó pode forçar Γ. Logo a afirmação "para todo o $k \in K$, $(k \models \Gamma \Rightarrow k \models F)$" é verdadeira. Portanto, $\Gamma \models F$.

($\forall^+$) Suponhamos que F foi obtida de uma fórmula φ por meio desta regra. Sendo assim, F é do tipo $\forall x \varphi(x)$. Por hipótese de indução, para todo o $k \in K$, $\forall b \in D(k)(k \models \Gamma \Rightarrow k \models \varphi(b))$[14]. Queremos provar que para todo o $k \in K$, $\forall b \in D(k)(k \models \Gamma \Rightarrow k \models \forall x \varphi(x))$.

Fixemos $k \in K$ tal que $k \models \Gamma$.

Seja $k' \geq k$ e c um elemento arbitrário de $D(k')$. Por monotonia, $k' \models \Gamma$.

Por hipótese de indução, $k' \models \Gamma \Rightarrow k' \models \varphi(c)$. Então $k' \models \varphi(c)$ e logo $k \models \forall x \varphi(x)$.

($\forall^-$) Por hipótese de indução:

Para todo o $k \in K, \forall b \in D(k)$ $(k \models \Gamma \Rightarrow k \models \forall x \varphi(x))$.

Partindo desta hipótese, é imediato que para todo o $k \in K$, $\forall b \in D(k)$ $(k \models \Gamma \Rightarrow k \models \varphi(b))$. Logo se $k \models \Gamma$ então $k \models F$.

---

[14]Pelo que vimos na 2.ª observação de 1.3.8 (pág. 19), b não pode ocorrer nas fórmulas de Γ (hipóteses da dedução).

($\exists^+$) Por hipótese de indução:

Para todo o $k \in K$, ($k \models \Gamma \Rightarrow k \models \varphi(a)$, para algum $a \in D(k)$).

Partindo desta hipótese, é imediato que para todo o $k \in K$, ($k \models \Gamma \Rightarrow k \models \exists x\varphi(x)$). Logo se $k \models \Gamma$ então $k \models F$.

($\exists^-$) Por hipótese de indução:

1. Para todo o $k \in K$, ($k \models \Gamma \Rightarrow k \models \exists x\varphi(x)$).

2. Para todo o $k \in K$, se $k \models \Gamma$ e $k \models \varphi(b)$, com $b \in D(k)$, então $k \models F$.

Fixemos $k \in K$ tal que $k \models \Gamma$. Por 1, $k \models \varphi(a)$ para algum $a \in D(k)$. Então $k \models \Gamma$ e $k \models \varphi(a)$. Logo, por 2, $k \models F$. ■

2.3.7 Observação

Como consequência directa do Teorema da validade temos que as leis clássicas $\neg\neg\varphi \rightarrow \varphi$, $(\varphi \vee \neg\varphi)$, $(\neg\varphi \vee \neg\neg\varphi)$, $\neg(\varphi \wedge \psi) \rightarrow (\neg\varphi \vee \neg\psi)$, $(\psi \rightarrow \varphi) \rightarrow (\neg\psi \vee \varphi)$, $\neg\forall x\varphi(x) \rightarrow \exists x\neg\varphi(x)$ e $(\varphi \rightarrow \exists x\psi(x)) \rightarrow \exists x(\varphi \rightarrow \psi(x))$, apresentadas no exemplos 1-6 desta secção como fórmulas que não são K-válidas, não são deriváveis na lógica intuicionista.

As definições que a seguir introduzimos vão ser-nos muito úteis para obtermos o Teorema de completude da semântica de Kripke relativamente à lógica intuicionista.

2.3.8 Definição

Seja Γ um conjunto de sentenças numa determinada linguagem $\mathcal{L}$. Γ é *consistente* se e só se $\Gamma \nvdash \perp$.

2.3.9 Definição

Seja Γ um conjunto de sentenças numa determinada linguagem $\mathcal{L}$ e $\mathcal{C}$ um conjunto de constantes dessa linguagem.

Γ é $\mathcal{C}$-Saturado se e só se:

(i) Γ é consistente

(ii) $\Gamma \vdash \varphi \Rightarrow \varphi \in \Gamma$ (ou seja, Γ é fechado para $\vdash$)

(iii) $\Gamma \vdash \varphi \vee \psi \Rightarrow \Gamma \vdash \varphi$ ou $\Gamma \vdash \psi$

(iv) $\Gamma \vdash \exists x\varphi(x) \Rightarrow$ existe $c \in \mathcal{C}$ tal que $\varphi(c) \in \Gamma$.

2.3.10 Observação

Poderíamos ter apresentado uma definição ligeiramente diferente substituindo as duas últimas alíneas por:

(iii) $\varphi \vee \psi \in \Gamma \Rightarrow \varphi \in \Gamma$ ou $\psi \in \Gamma$

(iv) $\exists x \varphi(x) \in \Gamma \Rightarrow$ existe $c \in \mathcal{C}$ tal que $\varphi(c) \in \Gamma$.

O Lema que a seguir demonstramos descreve como construir uma extensão $\mathcal{C}$-Saturada de um conjunto Γ de sentenças numa determinada linguagem $\mathcal{L}$.

2.3.11 Lema (Lema da Saturação)

Consideremos, numa determinada linguagem $\mathcal{L}$, um conjunto consistente de sentenças Γ e uma sentença φ qualquer.

Se $\Gamma \nvdash \varphi$ então existe Γ' $\mathcal{C}$-Saturado tal que $\Gamma' \supset \Gamma$ e $\Gamma' \nvdash \varphi$.

Demonstração

Comecemos por extender a linguagem $\mathcal{L}$, relativa a Γ, através de um conjunto numerável de constantes $\mathcal{C} = \{c_0, c_1, ..., c_n, ...\}$ que não pertençam a $\mathcal{L}$. Designemos por $\mathcal{L}'$ essa nova linguagem.

Vamos construir Γ' através de uma reunião de extensões de Γ tais que $\Gamma^0 \subseteq \Gamma^1 \subseteq \Gamma^2 ...$

Tomemos $\Gamma^0 = \Gamma$. Suponhamos que Γ^k é conhecido, contém apenas um número finito de novas constantes e $\Gamma^k \nvdash \varphi$.

Vamos dar uma definição indutiva das extensões Γ^k, considerando os casos k par ou k ímpar.

Caso 1. Se k é par, procuremos a primeira nova constante c_i que não pertece a Γ^k e a primeira[15] sentença existencial $\exists x \psi(x)$ não utilizada anteriormente, tal que $\Gamma^k \vdash \exists x \psi(x)$.

Definimos a nova extensão por $\Gamma^{k+1} := \Gamma^k \cup \psi(c_i)$.

Caso 2. Se k é ímpar, procuremos a primeira sentença disjuntiva $\psi_1 \vee \psi_2$ não utilizada anteriormente, tal que $\Gamma^k \vdash \psi_1 \vee \psi_2$.

Notemos que $\Gamma^k, \psi_1 \vdash \varphi$ e $\Gamma^k, \psi_2 \vdash \varphi$ não podem ocorrer simultaneamente, caso contrário teríamos $\Gamma^k \vdash \varphi$ por aplicação da regra $(\vee^-)$. Sendo assim, a nova extensão fica bem definida por $\Gamma^{k+1} := \begin{cases} \Gamma^k \cup \psi_1, \text{ se } \Gamma^k, \psi_1 \nvdash \varphi \\ \Gamma^k \cup \psi_2, \text{ se } \Gamma^k, \psi_2 \nvdash \varphi \end{cases}$

Seja então $\Gamma' = \bigcup\{\Gamma^k : k \in \mathbb{N}\}$.

É imediato que $\Gamma' \supset \Gamma$. Provemos que Γ' é $\mathcal{C}$-Saturado e que $\Gamma' \nvdash \varphi$.

Começamos por provar que Γ' é $\mathcal{C}$-Saturado.

(i) É imediato que Γ' é consistente, atendendo à forma como foi definido e ao facto de Γ ser consistente.

[15] Supomos fixada uma enumeração das fórmulas de $\mathcal{L}'$.

(ii) $\Gamma' \vdash \psi_1 \vee \psi_2 \Rightarrow \Gamma' \vdash \psi_1$ ou $\Gamma' \vdash \psi_2$

Seja k o menor número natural tal que $\Gamma^k \vdash \psi_1 \vee \psi_2$. Sendo assim, a sentença $\psi_1 \vee \psi_2$ não foi utilizada numa etapa anterior e pode ser utilizada na etapa k ou em qualquer etapa posterior a k, digamos l, uma vez que também aí se tem $\Gamma^l \vdash \psi_1 \vee \psi_2$. Então $\psi_1 \vee \psi_2$ vai ser utilizada numa etapa $h \geq k$. Logo, $\psi_1 \in \Gamma^{h+1}$ ou $\psi_2 \in \Gamma^{h+1}$ e, consequentemente, pela definição de Γ', $\psi_1 \in \Gamma'$ ou $\psi_2 \in \Gamma'$. É agora imediato que $\Gamma' \vdash \psi_1$ ou $\Gamma' \vdash \psi_2$.

(iii) $\Gamma' \vdash \psi_1 \Rightarrow \psi_1 \in \Gamma'$

Se $\Gamma' \vdash \psi_1$ então $\Gamma' \vdash \psi_1 \vee \psi_1$ e, pelo caso anterior, $\psi_1 \in \Gamma'$.

(iv) $\Gamma' \vdash \exists x\psi(x) \Rightarrow$ existe $c \in \mathcal{C}$ tal que $\psi(c) \in \Gamma'$.

Seja k o menor número natural tal que $\Gamma^k \vdash \exists x\psi(x)$. A sentença $\exists x\psi(x)$ irá ser utilizada numa etapa $h \geq k$. Logo, para algum $c_i \in \mathcal{C}$, $\psi(c_i) \in \Gamma^{h+1}$. Como $\Gamma^{h+1} \subseteq \Gamma'$ podemos concluir que existe $c \in \mathcal{C}$ tal que $\psi(c) \in \Gamma'$.

De (i), (ii), (iii) e (iv) concluímos que Γ' é $\mathcal{C}$-Saturado.

Resta verificar que $\Gamma' \nvdash \varphi$.

Provemos que $\Gamma^k \nvdash \varphi$, por indução em k.

Para $k = 0$, $\Gamma^k = \Gamma$ e, por hipótese, $\Gamma \nvdash \varphi$.

Verifiquemos o passo de indução considerando os casos k par ou k ímpar.

Consideremos que k é par e $\Gamma^k \nvdash \varphi$. Por definição, $\Gamma^{k+1} = \Gamma^k \cup \psi(c_i)$ onde c_i não pertece a Γ^k. Suponhamos que $\Gamma^{k+1} \vdash \varphi$, ou seja, $\Gamma^k \cup \psi(c_i) \vdash \varphi$. Nesse caso, como $\Gamma^k \vdash \exists x\psi(x)$ concluimos $\Gamma^k \vdash \varphi$ por $(\exists^-)$, o que contradiz a hipótese de indução. Logo $\Gamma^{k+1} \nvdash \varphi$.

Consideremos que k é ímpar e $\Gamma^k \nvdash \varphi$. Suponhamos que $\Gamma^{k+1} \vdash \varphi$. Então, por definição de Γ^{k+1}, vem $\Gamma^k \cup \psi_1 \vdash \varphi$, o que contradiz a suposição da definição $\Gamma^k, \psi_1 \nvdash \varphi$. Logo $\Gamma^{k+1} \nvdash \varphi$.

Acabámos de provar que, para qualquer k, $\Gamma^k \nvdash \varphi$.

Suponhamos então que $\Gamma' \vdash \varphi$. Então existe pelo menos um k tal que $\Gamma^k \vdash \varphi$ e obtemos uma contradição. Sendo assim, $\Gamma' \nvdash \varphi$. ■

2.3.12 Observação

Ao longo da demonstração anterior utilizámos frequentemente uma metalinguagem *clássica* (nomeadamente através de raciocínios por absurdo, onde está implícita a utilização do Princípio do Terceiro Excluído). Nos resultados que se seguem, essenciais para demonstrar a completude da semântica de Kripke, também iremos recorrer a essa metalinguagem, pelo que seria difícil convencer, por este processo, um verdadeiro intuicionista.

2.3.13 Teorema (Existência de modelo)

Numa determinada linguagem $\mathcal{L}$, consideremos um conjunto consistente de sentenças Γ e uma sentença φ qualquer.

Se $\Gamma \nvdash \varphi$ então existe um modelo de Kripke $\mathfrak{K}$, com primeiro nó k_0, tal que $k_0 \vDash \Gamma$ mas $k_0 \nvDash \varphi$.

Demonstração

Se $\Gamma \nvdash \varphi$ então, pelo Lema da Saturação, existe Γ' $\mathcal{C}$-Saturado tal que $\Gamma' \supset \Gamma$ e $\Gamma' \nvdash \varphi$. Seja $\mathcal{L}'$ a linguagem, com conjunto de constantes C, das sentenças de Γ'.

Consideremos uma sequência numerável C_1, ..., C_n, ... de conjuntos disjuntos e numeráveis de constantes que não pertencem a $\mathcal{L}'$.

Seja $C(\overrightarrow{n}) = C_0 \cup C_1 \cup ... \cup C_n$, para $n \geq 0$, onde $C_0 = C$.

Tomemos um modelo de Kripke $\mathfrak{K} \equiv (K, \leq, D, \vDash)$ tal que:

(a) K é o conjunto de todas as sequências finitas de números naturais, consideradas na sua ordem natural, ou seja, se $\overrightarrow{n}$ e $\overrightarrow{m}$ pertencem a K e $\overrightarrow{n} = (0, 1, 2, ...n)$ e $\overrightarrow{m} = (0, 1, 2, ...m)$ então $\overrightarrow{n} \leq \overrightarrow{m}$ sse $n \leq m$.

Refira-se que o nó k_0 é a sequência com apenas o número zero, que designaremos por $\overrightarrow{0}$. O conjunto de sentenças válidas nesse nó é Γ' e o conjunto de constantes correspondente é $C_0 = C$.

(b) $D(\overrightarrow{n}) = C(\overrightarrow{n})$ e $\mathcal{L}(\overrightarrow{n}) = \mathcal{L}' \cup C(\overrightarrow{n})$

(c) Seja $At(\overrightarrow{n})$ o conjunto de todas as fórmulas atómicas na linguagem $\mathcal{L}(\overrightarrow{n})$. Vamos definir o conjunto de fórmulas atómicas válidas em cada nó, $\Gamma(\overrightarrow{n})$, por indução no comprimento de $\overrightarrow{n}$.

Relembremos que o conjunto de sentenças válidas no primeiro nó é Γ' $\mathcal{C}$-Saturado. Tomemos então $\Gamma(\overrightarrow{0}) = \Gamma' \cap At(\overrightarrow{0})$.

Suponhamos que $\Gamma(\overrightarrow{n})$ está definido e vejamos como definir $\Gamma(\overrightarrow{n+1})$, por forma a ser $C(\overrightarrow{n+1})$-Saturado.

Consideremos uma enumeração de pares de sentenças em $\mathcal{L}(\overrightarrow{n})$, digamos $(\sigma_0, \tau_0), (\sigma_1, \tau_1), ..., (\sigma_i, \tau_i), ...$, tal que para cada $i \geq 0$, $\Gamma(\overrightarrow{n}) \cup \sigma_i \nvdash \tau_i$ (tomemos $\tau_i = \neg\sigma_i$, por exemplo).

Aplicando o Lema da Saturação a $\Gamma(\overrightarrow{n}) \cup \sigma_i$, para cada i, obtemos $\Gamma_i(\overrightarrow{n})$ saturado com respeito a uma nova linguagem $\mathcal{L}_i(\overrightarrow{n})$, verificando-se que $\sigma_i \in \Gamma_i(\overrightarrow{n})$ mas $\Gamma_i(\overrightarrow{n}) \nvdash \tau_i$. Seja $\Gamma_i(\overrightarrow{n+1}) = \Gamma_i(\overrightarrow{n}) \cap At(\overrightarrow{n})$.

Escolhemos então $\Gamma(\overrightarrow{n+1}) = \bigcup\limits_{i \geq 0} \Gamma_i(\overrightarrow{n+1})$ e $\mathcal{L}(\overrightarrow{n+1}) = \bigcup\limits_{i \geq 0} \mathcal{L}_i(\overrightarrow{n})$.

A saturação de $\Gamma(\overrightarrow{n+1})$, com respeito à linguagem $\mathcal{L}(\overrightarrow{n+1})$, resulta da saturação dos conjuntos $\Gamma_i(\overrightarrow{n+1})$, com respeito a $\mathcal{L}_i(\vec{n})$.

Provemos agora o seguinte facto:

2.3.14 Lema

Para todo o $\vec{n} \in K$, $\vec{n} \vDash \psi \Leftrightarrow \Gamma(\vec{n}) \vdash \psi$, qualquer que seja a sentença ψ na linguagem $\mathcal{L}(\vec{n})$.

Demonstração

Provemos o lema por indução na complexidade de ψ.

1. Se ψ for atómica a equivalência é verdadeira por definição de $\Gamma(\vec{n})$, já que $\vec{n} \vDash \psi \Leftrightarrow \psi \in \Gamma(\vec{n})$ e $\Gamma(\vec{n}) \vdash \psi \Rightarrow \psi \in \Gamma(\vec{n})$ porque, como vimos anteriormente, $\Gamma(\vec{n})$ é saturado (numa linguagem adequada). A implicação $\psi \in \Gamma(\vec{n}) \Rightarrow \Gamma(\vec{n}) \vdash \psi$ é óbvia.

2. Suponhamos agora que a equivalência é verificada por ψ e σ.

Caso $\psi \wedge \sigma$.

Sabemos que $\vec{n} \vDash \psi \wedge \sigma \Leftrightarrow \vec{n} \vDash \psi$ e $\vec{n} \vDash \sigma$. Por hipótese de indução, $\vec{n} \vDash \psi \Leftrightarrow \Gamma(\vec{n}) \vdash \psi$ e $\vec{n} \vDash \sigma \Leftrightarrow \Gamma(\vec{n}) \vdash \sigma$. Mas $\Gamma(\vec{n}) \vdash \psi$ e $\Gamma(\vec{n}) \vdash \sigma$ sse $\Gamma(\vec{n}) \vdash \psi \wedge \sigma$, por aplicação das regras da conjunção.

Logo, $\vec{n} \vDash \psi \wedge \sigma \Leftrightarrow \Gamma(\vec{n}) \vdash \psi \wedge \sigma$.

Caso $\psi \vee \sigma$. Demonstremos separadamente cada uma das implicações.

(a) $\vec{n} \vDash \psi \vee \sigma \Leftrightarrow \vec{n} \vDash \psi$ ou $\vec{n} \vDash \sigma$. Por hipótese de indução, $\Gamma(\vec{n}) \vdash \psi$ ou $\Gamma(\vec{n}) \vdash \sigma$. Logo, por aplicação de $(\vee^+)$, $\Gamma(\vec{n}) \vdash \psi \vee \sigma$.

(b) Recordemos que, na linguagem adequada, $\Gamma(\vec{n})$ é saturado. Logo, se $\Gamma(\vec{n}) \vdash \psi \vee \sigma$ então $\Gamma(\vec{n}) \vdash \psi$ ou $\Gamma(\vec{n}) \vdash \sigma$. Por hipótese de indução, $\vec{n} \vDash \psi$ ou $\vec{n} \vDash \sigma$ e, portanto $\vec{n} \vDash \psi \vee \sigma$.

Caso $\psi \to \sigma$.

(a) $\vec{n} \vDash \psi \to \sigma$. Suponhamos, com vista a um absurdo, que $\Gamma(\vec{n}) \nvdash \psi \to \sigma$. Então $\Gamma(\vec{n}), \psi \nvdash \sigma$. Pelo lema da Saturação, existe $\Gamma(\vec{m})$ saturado tal que $\Gamma(\vec{m}) \supseteq \Gamma(\vec{n}) \cup \psi$ e $\Gamma(\vec{m}) \nvdash \sigma$. Ora, se $\psi \in \Gamma(\vec{m})$ então $\Gamma(\vec{m}) \vdash \psi$ e, por hipótese de indução, $\vec{m} \vDash \psi$. Por outro lado, como $\vec{m} \geq \vec{n}$ e $\vec{n} \vDash \psi \to \sigma$ também $\vec{m} \vDash \psi \to \sigma$.

De $\vec{m} \vDash \psi \to \sigma$ e $\vec{m} \vDash \psi$ concluímos $\vec{m} \vDash \sigma$. Novamente por hipótese de indução, $\Gamma(\vec{m}) \vdash \sigma$. Contradição. Logo, $\Gamma(\vec{n}) \vdash \psi \to \sigma$.

(b) $\Gamma(\vec{n}) \vdash \psi \to \sigma$. Queremos provar que $\vec{n} \vDash \psi \to \sigma$. Consideremos $\vec{m} \geq \vec{n}$ tal que $\vec{m} \vDash \psi$. Por hipótese de indução, $\Gamma(\vec{m}) \vdash \psi$. Como $\Gamma(\vec{m}) \supseteq \Gamma(\vec{n})$, $\Gamma(\vec{m}) \vdash \psi \to \sigma$. Logo, por aplicação de $(\to^-)$, $\Gamma(\vec{m}) \vdash \sigma$ e, novamente por hipótese de indução, $\vec{m} \vDash \sigma$. Concluímos então que, para todo o $\vec{m} \geq \vec{n}$, se $\vec{m} \vDash \psi$ então $\vec{m} \vDash \sigma$, logo, $\vec{n} \vDash \psi \to \sigma$.

Caso $\exists x\psi(x)$.

(a) Se $\vec{n} \vDash \exists x\psi(x)$ então $\exists d \in D(\vec{n})(\vec{n} \vDash \psi(d))$. Pela hipótese de indução, $\Gamma(\vec{n}) \vdash \psi(d)$, para algum $d \in D(\vec{n})$, e, por aplicação de $(\exists^+)$, $\Gamma(\vec{n}) \vdash \exists x\psi(x)$.

(b) Pela saturação de $\Gamma(\vec{n})$, se $\Gamma(\vec{n}) \vdash \exists x\psi(x)$ então existe $c \in C(\vec{n})$ tal que $\psi(c) \in \Gamma(\vec{n})$. Ora, se $\psi(c) \in \Gamma(\vec{n})$ então $\vec{n} \vDash \psi(c)$, com $c \in D(\vec{n})$ e, portanto, $\vec{n} \vDash \exists x\psi(x)$.

Caso $\forall x\psi(x)$.

(a) Se $\vec{n} \vDash \forall x\psi(x)$ então para todo o $\vec{m} \geq \vec{n}$ e $c \in D(\vec{m})$, $\vec{m} \vDash \psi(c)$. Por hipótese de indução, $\Gamma(\vec{m}) \vdash \psi(c)$. Suponhamos que $\Gamma(\vec{n}) \nvdash \forall x\psi(x)$. Seja a uma constante pertencente a uma linguagem de um nó superior a $\vec{n}$, mas não pertencente a $\mathcal{L}(\vec{n})$, ou seja, $a \in C^{n+1}$, tal que $\Gamma(\vec{n}) \nvdash \psi(a)$. Pelo lema da Saturação, existe $\Gamma(\vec{m})$ saturado tal que $\Gamma(\vec{m}) \supseteq \Gamma(\vec{n})$ e $\Gamma(\vec{m}) \nvdash \psi(a)$. Contradição. Logo, $\Gamma(\vec{n}) \vdash \forall x\psi(x)$.

(b) Se $\Gamma(\vec{n}) \vdash \forall x\psi(x)$ então $\Gamma(\vec{n}) \vdash \psi(c)$, qualquer que seja c, por aplicação de $(\forall^-)$. Por hipótese de indução, $\vec{n} \vDash \psi(c)$, qualquer que seja c, e, portanto, $\vec{n} \vDash \forall x\psi(x)$.

c.q.d

Finalizámos a demonstração do Lema 2.3.14. Tendo em conta este facto torna-se fácil terminar a demonstração do teorema da existência.

Ao longo desta demonstração construímos um modelo $\mathfrak{K}$ cujo primeiro nó, $\vec{0}$, força Γ, visto que $\vec{0} \vDash \Gamma'$ e $\Gamma \subset \Gamma'$.

Por outro lado, $\Gamma' \nvdash \varphi$ implica que $\Gamma(\vec{0}) \nvdash \varphi$. Portanto, pelo lema que acabámos de provar, $\vec{0} \nvDash \varphi$.

Assim, como $k_0 = \vec{0}$, temos que $k_0 \vDash \Gamma$ mas $k_0 \nvDash \varphi$. ■

Desta forma reunimos todas as condições para provar o teorema que nos propusemos obter no início desta secção.

2.3.15 Teorema (Completude)

Seja Γ um conjunto de fórmulas fechado e F uma sentença. Então $\Gamma \vDash F \Rightarrow \Gamma \vdash F$. Ou seja, se F é uma Kripke-consequência de um conjunto Γ, então F é derivável de Γ.

Em particular, se $\vDash F$ então $\vdash F$. Ou seja, se F é K-válida, então F é uma lei dedutível na lógica intuicionista.

Demonstração

Partindo da hipótese $\Gamma \models F$, suponhamos, com vista a um absurdo, que $\Gamma \nvdash F$.

Pelo teorema anterior, existe um modelo de Kripke $\mathfrak{K}$, com primeiro nó k_0, tal que $k_0 \models \Gamma$ mas $k_0 \not\models F$. Portanto, $\mathfrak{K} \models \Gamma$ mas $\mathfrak{K} \not\models F$, logo $\Gamma \not\models F$, o que contradiz a hipótese inicial. Concluimos então que $\Gamma \vdash F$. ■

Vimos anteriormente como a simplicidade desta semântica permitiu refutar alguns resultados clássicos. Para finalizar este capítulo pretendemos apresentar ainda mais duas aplicações quase imediatas desta semântica.

Recordemos as propriedades intuicionistas da disjunção e do quantificador existencial referidas na secção 1.1.

2.3.16 Definição

Um conjunto de sentenças Γ verifica a:

(a) **Propriedade da disjunção** se: $\Gamma \vdash \varphi \vee \psi \Rightarrow \Gamma \vdash \varphi$ ou $\Gamma \vdash \psi$.

(b) **Propriedade da existência** se: $\Gamma \vdash \exists x\varphi(x) \Rightarrow \Gamma \vdash \varphi(a)$.

(Sempre que em $\varphi \vee \psi$ e $\exists x\varphi(x)$ não ocorrem variáveis livres)

Referimos já que a lógica clássica não admite as propriedades da disjunção e da existência. Provemos que a lógica intuicionista (sem símbolos funcionais) goza destas propriedades.

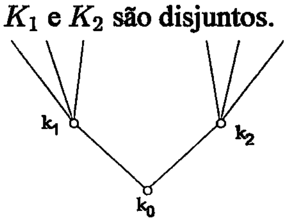
2.3.17 Teorema (Propriedade da disjunção)

A lógica intuicionista, sem símbolos funcionais, goza da propriedade da disjunção.

Demonstração

Suponhamos, com vista a um absurdo, que $\vdash \varphi \vee \psi$ mas $\nvdash \varphi$ e $\nvdash \psi$.

Então existem modelos de Kripke $\mathfrak{K}_1$ e $\mathfrak{K}_2$ com primeiros nós k_1 e k_2, respectivamente, tais que $k_1 \not\models \varphi$ e $k_2 \not\models \psi$. Sem perda de generalidade suponhamos que os respectivos conjuntos parcialmente ordenados K_1 e K_2 são disjuntos.



Seja $\mathfrak{K}$ o modelo de Kripke representado na figura anterior, com a linguagem conjunta de $\mathfrak{K}_1$ e $\mathfrak{K}_2$, digamos $\mathcal{L}$, com $K = K_1 \cup K_2 \cup \{k_0\}$ sendo k_0 um nó que não pertence a $\mathfrak{K}_1$ nem a $\mathfrak{K}_2$, ou seja, $k_0 \notin K_1 \cup K_2$.

Seja $\Gamma(k)$ o conjunto de sentenças verdadeiras no nó $k \in K$.

Tomemos $\Gamma(k) = \begin{cases} \Gamma_1(k), \text{se } k \in K_1 \\ \Gamma_2(k), \text{se } k \in K_2 \\ C, \text{se } k = k_0 \end{cases}$

Onde C é o conjunto de constantes da linguagem $\mathcal{L}$.

Deste modo $\mathfrak{K}_1$ e $\mathfrak{K}_2$ são sub-modelos de $\mathfrak{K}$, na medida em que a relação binária $\models$ induzida em $\mathfrak{K}_1$, por meio de $\mathfrak{K}$, coincide com a sua antiga relação $\models$ **(van Dalen, 1983)**.

Por hipótese, $\vdash \varphi \vee \psi$, logo pelo teorema de completude, $k_0 \models \varphi \vee \psi$. Por definição $(\mathbf{K_3})$, $k_0 \models \varphi$ ou $k_0 \models \psi$. Mas se $k_0 \models \varphi$ então, por monotonia, $k_1 \models \varphi$ (contradição) e se $k_0 \models \psi$ então, por monotonia $k_2 \models \psi$ (contradição).

Logo $\vdash \varphi$ ou $\vdash \psi$. ■

2.3.18 Observação

A demonstração da propriedade da disjunção no caso da lógica proposicional intuicionista seria ainda mais simples. Bastaria colocar um novo nó k_0 (abaixo de k_1 e k_2) em que nenhuma fórmula atómica fosse forçada.

2.3.19 Teorema (Propriedade da existência)

A lógica intuicionista, com uma linguagem sem símbolos funcionais e com pelo menos uma constante, goza da propriedade da existência.

Demonstração

Suponhamos, com vista a um absurdo, que $\vdash \exists x\varphi(x)$ mas $\nvdash \varphi(a)$, qualquer que seja a constante a.

Para cada constante c existe um modelo de Kripke $\mathfrak{K}_c$, com primeiro nó k_c, tal que $k_c \nVdash \varphi(c)$. Utilizando um argumento análogo ao da demonstração anterior, construimos um modelo de Kripke $\mathfrak{K}$ tomando a união disjunta de todos os $\mathfrak{K}_c$ e adicionando um primeiro nó k_0.

Por hipótese, $\vdash \exists x\varphi(x)$, logo pelo teorema de completude, $k_0 \models \exists x\varphi(x)$. Por definição, existe uma constante a tal que $k_0 \models \varphi(a)$. Por monotonia, todos os outros nós forçam $\varphi(a)$, o que contradiz o facto de existir um k_a tal que $k_a \nVdash \varphi(a)$. ■

Capítulo 3

Semântica de demonstrabilidade de Artemov

3.1 A semântica desejada

Sergei Artemov, no seu artigo "Explicit provability: the inteded semantics for intuitionistic and modal logic" apresentou uma solução para o problema, há muito em aberto, de formalizar a semântica de demonstrabilidade definida implicitamente por Brouwer e estabelecer a completude da lógica proposicional intuicionista com respeito a essa semântica (**Artemov, 1998**). O ponto de partida, que constitui o alicerce de todo o trabalho desenvolvido, é o sistema formal $\mathcal{LP}$ - *Logic of Proofs* - que interioriza as provas como termos.

Neste capítulo vamos explorar o sistema $\mathcal{LP}$, apresentar a semântica desejada por Gödel quando esboçou a sua lógica de demonstrabilidade explícita em 1938 e demonstrar a validade e completude de $\mathcal{LP}$ com respeito a essa semântica.

Pelas limitações de espaço e por ser necessário ainda uma breve apresentação da lógica modal S4 de Gödel para demonstrar que $\mathcal{LP}$ realiza S4, não nos será possível alongar este trabalho por forma a obter a formalização de BHK e estabelecer a completude da lógica proposicional intuicionista com respeito a essa semântica. Limitamo-nos a fazer um esboço de todo o processo na secção prospectiva 3.6.

3.2 O Sistema $\mathcal{LP}$

Antes de apresentar o conjunto de axiomas e regras de inferência em que se baseia o sistema formal $\mathcal{LP}$, vamos identificar a sua linguagem.

A linguagem de $\mathcal{LP}$ inclui:

- A linguagem usual da lógica proposicional clássica;
- Variáveis prova $u, v, w, x, y, z, \ldots$ (possivelmente com índices);
- Constantes prova $a, b, c, \ldots$ (possivelmente com índices);
- Símbolos funcionais: símbolo 0-ário !, símbolos binários $.$ e $+$;
- Símbolo predicativo do tipo "termo : fórmula".

Vamos utilizar $i, j, k, l, m, n, \ldots$ para designar números naturais.

3.2.1 Definição

As variáveis e as constantes são **termos**.

São também termos as expressões da forma $!t$, $t_1.t_2$ e $t_1 + t_2$, onde t, t_1 e t_2 são termos.

Nada mais é termo.

3.2.2 Observação

Os termos definidos anteriormente serão designados por **polinómios prova** e notados por p, r, s, t.... Analogamente iremos utilizar a designação **coeficientes** para nos referirmos às constantes prova, que não serão mais do que provas de factos básicos, nomeadamente, de axiomas proposicionais.

3.2.3 Definição

Seja t um termo qualquer e S uma letra proposicional.

As regras de formação de **fórmulas** são as seguintes:

1. Uma proposição S é fórmula.
2. Se t é um termo e A é uma fórmula, então $t : A$ é uma fórmula.
3. Se A e B são fórmulas, então $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ são fórmulas.

Nada mais é fórmula .

3.2.4 Observação

A interpretação da fórmula $t : A$ é "t é uma prova de A". O mesmo termo t pode ser prova de várias fórmulas.

3.2.5 Notação

Nesta secção utilizamos:

$A, B, C, E, F, G, H, X, Y, Z, ...$ para fórmulas;

$\Gamma, \Delta, ...$ para conjuntos finitos de fórmulas;

$\vec{x}, \vec{y}, \vec{z}, ...$ para vectores de variáveis prova;

$\vec{p}, \vec{r}, \vec{s}, ...$ para vectores de polinómios prova;

$\#E$ para o número de Gödel de E.[16]

Sendo $\vec{s} = (s_1, ..., s_n)$ e $\Gamma = (F_1, ..., F_n)$ escrevemos:

$\vec{s} : \Gamma$ como abreviatura de $(s_1 : F_1, ..., s_n : F_n)$;

$\bigvee\Gamma$ como abreviatura de $F_1 \vee ... \vee F_n$

$\bigwedge\Gamma$ como abreviatura de $F_1 \wedge ... \wedge F_n$

[16] O processo de godelização (ou codificação) consiste em associar a cada símbolo, expressão ou sequência finita de expressões, de $\mathcal{L}_{ar}$ (secção 3.3), um número natural.

Aos símbolos da lista $\neg, \wedge, \vee, \rightarrow, (,), \forall, \exists, =, 0, ', +, \times, \uparrow, <$ associamos, respectivamente, os números $1, 2, 3...15$ e às variáveis x_i (com $i = 0, 1, ...$) os números $16 + 2i$. O número associado a cada símbolo s é notado por $\#s$ e designado por código de s ou número de Gödel de s.

Para extender a codificação aos termos e fórmulas, utilizamos a unicidade da decomposição em factores primos, associando a cada expressão $s_1 s_2 ... s_k$ o código $2^{\#s_1} \times 3^{\#s_2} \times ... \times p_k^{\#s_k}$, onde p_k é o $k+1$-ésimo número primo.

Finalmente, o processo é concluído associando a cada dedução $\xi_1, \xi_2, ..., \xi_n$ (sequência finita de expressões) o código $3^{\#\xi_1} \times 5^{\#\xi_2} ... \times p_{n+1}^{\#\xi_n}$.

Vamos assumir que os símbolos funcionais têm prioridade em relação aos conectivos, estabelecendo-se a seguinte ordem, começando com o símbolo prioritário:

$!, ., +, :, \neg, \wedge, \vee, \rightarrow$

3.2.6 Definição

Sistema $\mathcal{LP}$

Indicamos de seguida os Axiomas e Regras em que se baseia o Sistema $\mathcal{LP}$.

Axiomas:

A_0) Número finito de axiomas-esquema[17] da lógica clássica proposicional, na linguagem de $\mathcal{LP}$.

A_1) $t : F \rightarrow F$ (Axioma de **verificação**)

A_2) $t : (F \rightarrow G) \rightarrow (s : F \rightarrow (t.s) : G)$ (Axioma de **aplicação**)

A_3) $t : F \rightarrow !t : (t : F)$ (Axioma do **verificador de provas**)

A_4) $t : F \rightarrow (s + t) : F,\ s : F \rightarrow (s + t) : F$ (Axioma de **escolha**)

Regras de inferência:

R_1) $\frac{F \rightarrow G \quad F}{G}$ **(Modus Ponens)**

R_2) $\frac{}{c:A}$ sendo A um axioma do tipo A_0-A_4 **(Necessidade de Axioma)**

3.2.7 Observações

1. O axioma A_3 traduz formalmente a interpretação informal do símbolo !: instrumento através do qual um observador exterior à teoria verifica que, efectivamente, t é uma prova de F.

2. Se retirarmos a regra R_2 ao sistema $\mathcal{LP}$, obtemos o sistema $\mathcal{LP}_0$.

3. Notemos que não há restrições para a escolha de c na regra R_2, ou seja, dentro de uma derivação, o mesmo coeficiente pode ser utilizado como prova de diferentes axiomas.

[17]Um axioma-esquema, ou esquema de axiomas, é um conjunto (normalmente infinito) de fórmulas, sendo cada uma destas um axioma. O mais comum é este conjunto de axiomas ser construído recursivamente. Refira-se a título de exemplo, o axioma de indução da aritmética de Peano.
No entanto, certos autores, como Artemov, designam também por axioma-esquema cada um dos axiomas seguintes, correspondentes a algumas regras de inferência (Dummett, 2000):
$A \rightarrow (B \rightarrow A)$; $A \rightarrow (B \rightarrow A \wedge B)$; $A \wedge B \rightarrow A$; $A \wedge B \rightarrow B$; $A \rightarrow A \vee B$; $B \rightarrow A \vee B$; $A \vee B \rightarrow ((A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow C))$; $(A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$; $(A \rightarrow B) \rightarrow ((A \rightarrow \neg B) \rightarrow \neg A)$; $A \rightarrow (\neg A \rightarrow B)$; $\forall x A(x) \rightarrow A(t)$; $A(t) \rightarrow \exists x A(x)$
O primeiro axioma-esquema $A \rightarrow (B \rightarrow A)$, por exemplo, corresponde a uma infinidade de axiomas particulares, um para cada fórmula A, afirmando que, se A é verdadeira então $B \rightarrow A$ também o é.

3.2.8 Definição

Uma especificação de constantes, EC, é um conjunto finito de fórmulas do tipo $c_i : A_i$, em que c_i é uma constante prova e A_i é um dos axiomas A_0-A_4.

Pela definição anterior, cada derivação de $\mathcal{LP}$ gera uma EC formada pelo conjunto de fórmulas introduzidas por meio da regra *necessidade de axioma*, R_2.

Vejamos alguns exemplos representativos de derivações de $\mathcal{LP}$ e a forma como este sistema formaliza a interpretação BHK.

3.2.9 Exemplos:

1.

$$1 \quad t : (A \rightarrow B) \rightarrow (x : A \rightarrow (t.x) : B) \quad (A_2)$$

Esta derivação, que consiste apenas no axioma A_2, traduz o facto de uma demonstração de $A \rightarrow B$ ser, segundo a interpretação BHK, uma prova que transforma uma demonstração x de A numa demonstração tx de B.

2.

1	$A \rightarrow (B \rightarrow A \wedge B)$	(A_0)
2	$c : (A \rightarrow (B \rightarrow A \wedge B))$	1 (R_2)
3	$c : (A \rightarrow (B \rightarrow A \wedge B)) \rightarrow (x : A \rightarrow (c.x) : (B \rightarrow A \wedge B))$	(A_2)
4	$x : A \rightarrow (c.x) : (B \rightarrow A \wedge B)$	2, 3 (MP)
5	$(c.x) : (B \rightarrow A \wedge B) \rightarrow (y : B \rightarrow (c.x.y) : A \wedge B)$	(A_2)
6	$x : A \rightarrow (y : B \rightarrow (c.x.y) : A \wedge B)$	4, 5 (lógica proposicional)
7	$x : A \wedge y : B \rightarrow (c.x.y) : A \wedge B$	6 (lógica proposicional)

Se temos uma demonstração de A e uma demonstração de B, então podemos obter uma demonstração para $A \wedge B$.

3.

1	$A \rightarrow A \vee B,\ B \rightarrow A \vee B$	(A_0)
2	$a : (A \rightarrow A \vee B),\ b : (B \rightarrow A \vee B)$	1 (R_2)
3	$x : A \rightarrow (a.x) : (A \vee B),\ y : B \rightarrow (b.y) : (A \vee B)$	(A_2 e lógica proposicional)
4	$(a.x) : (A \vee B) \rightarrow (a.x + b.y) : (A \vee B)$, $(b.y) : (A \vee B) \rightarrow (a.x + b.y) : (A \vee B)$	(A_4)
5	$x : A \rightarrow (a.x + b.y) : (A \vee B), y : B \rightarrow (a.x + b.y) : (A \vee B)$	3, 4 (lógica proposicional)
6	$(x : A \vee y : B) \rightarrow (a.x + b.y) : (A \vee B)$	5 (lógica proposicional)

Se temos uma de duas hipóteses, ou uma demonstração de A ou uma demonstração de B, então temos uma demonstração para $A \vee B$.

3.2.10 Definição

Uma **derivação ou dedução**, em $\mathcal{LP}$, de uma fórmula A a partir de um conjunto (finito) de fórmulas Γ, é uma sequência finita de fórmulas, em que cada uma delas é um elemento de Γ (hipótese da dedução), ou uma ocorrência de um dos axiomas A_0-A_4, ou é inferida por meio da regra R_1 a partir de duas fórmulas precedentes, ou é introduzida a partir de R_2.

3.2.11 Notação

$\Gamma \vdash A$ significa que existe uma derivação de A a partir do conjunto Γ.

Deveríamos alterar a notação usual $\vdash$ para, por exemplo, $\vdash_{\mathcal{LP}}$. No entanto, quando o contexto não deixar dúvidas acerca do sistema subjacente, continuaremos a utilizar o símbolo $\vdash$.

3.2.12 Teorema da Dedução

Sejam A e B fórmulas. Para todo o conjunto Γ, se $\Gamma, A \vdash B$ então $\Gamma \vdash A \to B$.

Demonstração

Suponhamos que existe uma derivação de B com hipóteses Γ, A. Estudemos todos os casos possíveis para introdução da fórmula B.

1) $B \in \Gamma$

Consideremos a seguinte sub-derivação:

$$\begin{array}{lll} n & B & \\ n+1 & B \to (A \to B) & (A_0) \\ n+2 & A \to B & 1,2\,(R_1) \end{array}$$

Se $B \in \Gamma$ então, obviamente, $\Gamma \vdash B$. Acrescentando a essa derivação, as três linhas anteriores, obtemos $\Gamma \vdash A \to B$.

2) $B = A$

Por A_0, $\vdash A \to A$. Logo $\Gamma \vdash A \to A$.

3) Suponhamos agora que B é um dos axiomas $A_0 - A_4$. Podemos obter a seguinte derivação sem hipóteses:

$$\begin{array}{lll} 1 & a : B & (R_2) \\ 2 & B \to (A \to B) & (A_0) \\ 3 & b : \big(B \to (A \to B)\big) & (R_2) \\ 4 & ba : A \to B & (A_2 \text{ e lógica proposicional}) \\ 5 & (ba : A \to B) \to (A \to B) & (A_1) \\ 6 & A \to B & 4,5\,(R_1) \end{array}$$

Sendo assim $\vdash A \to B$ e, portanto, $\Gamma \vdash A \to B$.

4) Suponhamos que, na derivação $\Gamma, A \vdash B$, B é introduzido por meio de R_1 a partir de, digamos, $G \to B$ e G. Estudemos os seguintes sub-casos:

Se $G \to B$, $G \in \Gamma$ então $\Gamma \vdash B$ por R_1. Acrescentando a esta última derivação a sub-derivação do caso 1) obtemos $\Gamma \vdash A \to B$.

Se $G \to B \in \Gamma$ e $G = A$ então temos $A \to B \in \Gamma$ e obviamente $\Gamma \vdash A \to B$.

Se $G \to B = A$ e $G \in \Gamma$ então $\Gamma \vdash G$ (pois $G \in \Gamma$). Continuando esta derivação $\mathcal{D}$ obtemos

1	Γ	H
...	$\mathcal{D}$	
n	G	
$n+1$	$G \to ((G \to B) \to B)$	(A_0)
5	$(G \to B) \to B$	$n, n+1$ (R_1)

Ou seja, $\Gamma \vdash A \to B$.

5) Suponhamos, finalmente, que B é introduzido, na derivação $\Gamma, A \vdash B$, por meio de R_2, ou seja B é do tipo $c : F$ em que F é um dos axiomas $A_0 - A_4$. Consideremos a seguinte derivação sem hipóteses:

1	B	(R_2)
2	$B \to (A \to B)$	(A_0)
3	$(A \to B)$	$1, 2$ (R_1)

Sendo assim $\vdash A \to B$ e, portanto, $\Gamma \vdash A \to B$. ■

3.2.13 Observação

A implicação do Teorema anterior é na verdade uma equivalência, ou seja, $\Gamma, A \vdash B$ sse $\Gamma \vdash A \to B$.

Se $\Gamma \vdash A \to B$ então $\Gamma, A \vdash A \to B$. Como é óbvio $\Gamma, A \vdash A$. Logo, por aplicação da regra Modus Ponens, $\Gamma, A \vdash B$.

3.2.14 Notação

Utilizamos $\mathcal{LP}$(EC) para designar $\mathcal{LP}_0$ com a especificação de constantes EC.

3.2.15 Proposição (Lema do Transporte)

Seja $\mathcal{D}$ uma derivação do tipo $\vec{x} : \Gamma, \Delta \vdash F$. Então existe uma prova polinomial $t(\vec{x}, \vec{y})$ tal que: se $\vec{y} : \Delta$ então $\vec{x} : \Gamma, \vec{y} : \Delta \vdash t(\vec{x}, \vec{y}) : F$.

Demonstração

Demonstremos a proposição por indução na derivação $\mathcal{D}$.

1) Caso em que $F \in \vec{x} : \Gamma$ ou $F \in \Delta$.

Se $F = x_i : A_i \in \vec{x} : \Gamma$ tomamos $t(\vec{x}, \vec{y}) = !x_i$ já que $x_i : A_i \to !x_i : (x_i : A_i)$, por A_3.

Se $F = B_j \in \Delta$ tomamos $t(\vec{x}, \vec{y}) = y_j$.

2) Casos em que F é um dos axiomas ou uma das regras de $\mathcal{LP}$.

Se F é um dos axiomas A_0 - A_4 escolhemos um coeficiente c que ainda não tenha sido utilizado e tomamos $t(\vec{x}, \vec{y}) = c$. Por R_2 $\vdash t(\vec{x}, \vec{y}) : F$.

Suponhamos que F é introduzida por meio de R_1, a partir de $G \rightarrow F$ e de G. Por hipótese de indução, existem $u(\vec{x}, \vec{y})$ e $v(\vec{x}, \vec{y})$ tais que:

$$\vec{x} : \Gamma, \vec{y} : \Delta \vdash u(\vec{x}, \vec{y}) : G \rightarrow F \text{ e } \vec{x} : \Gamma, \vec{y} : \Delta \vdash v(\vec{x}, \vec{y}) : G.$$

Por aplicação de A_2, $\vec{x} : \Gamma, \vec{y} : \Delta \vdash u.v : F$, portanto escolhemos $t(\vec{x}, \vec{y}) = u(\vec{x}, \vec{y}).v(\vec{x}, \vec{y})$.

Suponhamos que F é introduzida por meio de R_2. Então F é do tipo $a : A$, para alguma prova constante a e axioma A. Por hipótese, $\vec{x} : \Gamma, \Delta \vdash a : A$ e por A_2, $a : A \rightarrow !a : (a : A)$. Logo $\vec{x} : \Gamma, \vec{y} : \Delta \vdash !a : (a : A)$ e portanto escolhemos $t = !a$. ■

3.2.16 Nota: Em particular, se $\Delta \vdash F$ podemos construir uma prova polinomial $t(\vec{y})$ que é produto de constantes e variáveis de uma prova $\vec{y}$ de Δ, tal que $\vec{y} : \Delta \vdash t(\vec{y}) : F$.

3.2.17 Corolário (Regra da necessidade)

Se $\vdash F$ então existe uma prova polinomial p tal que $p : F$.

Demonstração

Este corolário é uma consequência directa do teorema anterior. Pela demonstração do teorema temos que:

Se F é um axioma, p é uma prova constante.

Se F é introduzida por Modus Ponens a partir de dois axiomas, p é um produto de duas constantes.

Se F é introduzida por R_2, p é uma prova do tipo $!c$ onde c é uma prova constante.

Podemos então concluir que a prova p existe e é um polinómio sem variáveis, apenas com coeficientes, e sem a função $+$. ■

3.3 Interpretação associada a $\mathcal{LP}$

O papel desempenhado pelo sistema $\mathcal{LP}$ com respeito à noção de prova é idêntico ao papel desempenhado pelas funções booleanas da lógica proposicional para a noção de sentença (**Artemov, 1998**). Tal como as funções booleanas permitem obter o teorema da completude da lógica proposicional clássica, também $\mathcal{LP}$ irá permitir obter a completude da lógica proposicional intuicionista.

Pelo que vimos na secção anterior, no sistema $\mathcal{LP}$ as provas são também termos. Um sistema com um operador verificador de provas, que admita as próprias provas como termos, em particular, um sistema para a aritmética de Peano **AP**, constitui um modelo para $\mathcal{LP}$. Neste caso, os números de Gödel são o instrumento que permite interiorizar as provas como termos. A aritmética de Peano, não sendo necessária para demonstrar a validade da semântica $\mathcal{LP}$, vai ser fundamental no nosso trabalho para estabelecer o Teorema da Completude.

Comecemos por identificar a linguagem de **AP**, $\mathcal{L}_{ar}$.

A linguagem $\mathcal{L}_{ar}$ inclui os seguintes símbolos não lógicos primitivos:

- Uma constante: 0 (zero)
- Três símbolos funcionais binários: $+$ (adição), $\times$ (multiplicação) e $\uparrow$ (exponenciação);
- Um símbolo funcional unário: $'$ (sucessor)
- Um símbolo relacional binário: $<$

Embora o símbolo $<$ possa ser definido a partir de $+$, 0 e o símbolo $\uparrow$ possa ser introduzido a partir de $\times$, optamos pela inclusão destes dois símbolos nos símbolos primitivos, por uma questão de simplificação de escrita.

A igualdade (formalmente um símbolo predicativo ou relacional) será considerada como um símbolo lógico.

3.3.1 Definição

A constante 0 é um **termo** e $0', 0'', \ldots$ são termos (usualmente designados por numerais).

As variáveis aritméticas são termos.

São também termos as expressões da forma $t_1 \times t_2$, $t_1 \uparrow t_2$ e $t_1 + t_2$, onde t_1 e t_2 são termos.

Nada mais é termo.

3.3.2 Definição

1. Consideremos os termos t_1 e t_2. Então t_1, t_2, $t_1 < t_2$ e $t_1 = t_2$ são fórmulas.

2. Se A e B são fórmulas, então $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ $\forall x \varphi(x)$ e $\exists x \varphi(x)$ são fórmulas.

3. Nada mais é **fórmula**.

As fórmulas do tipo 1 dizem-se atómicas.

3.3.3 Notação

Nesta secção utilizamos:

φ, ψ... para fórmulas aritméticas;

f, g, h... para termos aritméticos;

$u, v, w, x, y, z, \ldots$ para variáveis aritméticas[18];

Δ_1 e Σ_1 são classes de predicados da aritmética (a definir);

Continuaremos a utilizar $i, j, k, l, m, n, \ldots$ para designar números naturais. Se n é um número natural, $\overline{n}$ designa o numeral correspondente.

Conforme **Oliveira (1996)**, os axiomas da aritmética de Peano (teoria **AP** na linguagem $\mathcal{L}_{ar}$) são os axiomas $\mathrm{AP}_1 - \mathrm{AP}_{10}$ e o axioma de indução:

AP_1. $\forall x(x' \neq 0)$

AP_2. $\forall xy(x' = y' \rightarrow x = y)$

AP_3. $\forall x(x + 0 = x)$

AP_4. $\forall xy\big(x + y' = (x + y)'\big)$

AP_5. $\forall x(x \times 0 = 0)$

AP_6. $\forall x\big(x \times y' = x \times y + x\big)$

AP_7. $\forall x\big(x \uparrow 0 = 0'\big)$

AP_8. $\forall xy\big(x \uparrow y' = x \uparrow y \times x\big)$

Axiomas de ordem:

AP_9. $\forall x\big(x \not< 0\big)$

AP_{10}. $\forall xy\big(x < y' \leftrightarrow x \leq y\big)$

Axioma de indução:

Axioma-esquema (Ind) $\forall \overline{y}\Big(\varphi(0, \overline{y}) \wedge \forall x\big(\varphi(x, \overline{y}) \rightarrow \varphi(x', \overline{y})\big) \rightarrow \forall x \varphi(x, \overline{y})\Big)$

3.3.4 Observação

Na axiomática apresentada utilizámos algumas abreviaturas, nomeadamente, $t_1 \neq t_2$ para $\neg(t_1 = t_2)$, $t_1 \not< t_2$ para $\neg(t_1 < t_2)$ e $t_1 \leq t_2$ para $(t_1 < t_2) \vee (t_1 = t_2)$.

Torna-se necessário neste momento fazer uma breve discussão da semântica normalmente associada a **AP**.

Cada termo fechado designa um único número natural (0 designa **0**, $0'$ designa 1, $0' \times 0''$ designa 2...). Se t_1 e t_2 denotam os números naturais i e j, t_1' designa $i + 1$, $t_1 \times t_2$ designa ij, etc.

---

[18]Utilizamos a mesma notação para variáveis aritméticas de **AP** e variáveis prova de $\mathcal{LP}$, certos de que, em cada caso, o contexto nos permite distinguir o tipo de variável a que nos referimos.

Uma fórmula de **AP** diz-se *verdadeira* se for verdadeira[19] quando as suas variáveis tomam valores nos elementos do conjunto $\mathbb{N}$, da axiomática Zermelo-Fraenkel, $0, 1, 2...$ A título de exemplo, a fórmula $x + y = y + x$ é verdadeira.

A fórmula $\exists x F(x)$ é verdadeira se existir um número natural i tal que o resultado da substituição em F de x por i é verdadeiro (teorema de ZF válido em $\mathbb{N}$).

Antes de prosseguirmos, e com o intuito de chegar à interpretação desejada da semântica $\mathcal{LP}$, vamos deter-nos numa breve revisão do conceito de recursividade.

Comecemos por recordar o conceito de recursividade primitiva.

Seja $k > 1$ um número natural. Consideremos g e r, duas funções de domínio $\mathbb{N}^{k-1}$ e $\mathbb{N}^{k+1}$, respectivamente. Diz-se que uma função f de domínio $\mathbb{N}^k$ é definida por **recursividade primitiva**, a partir de g e r, sse para cada um dos seus argumentos se verifica o seguinte esquema recursivo **(Felscher, 2000)**:

$f(0, \overrightarrow{x}) = g(\overrightarrow{x})$

$f(n+1, \overrightarrow{x}) = r\big(f(n, x), n, \overrightarrow{x}\big)$ para todo o $\overrightarrow{x} \in \mathbb{N}^{k-1}$ e $n \geq 0$.

3.3.5 Definição

Uma classe **F** de funções diz-se **fechada para a recorrência primitiva** (abreviadamente FRP) sse:

- **F** contém as funções recursivas primitivas iniciais:

(i) S ou $'$ (sucessor)

(ii) C_0 (função constante nula, ou seja, $C_0(n) = 0$ para todo $n \in \mathbb{N}$)

(iii) p_i^k(projecções, $p_i^k(\overrightarrow{x}) = x_i$ onde $\overrightarrow{x} \in \mathbb{N}^k$)

- **F** é fechada para:

(iv) a superposição[20]

(v) a recursividade primitiva.

3.3.6 Definição

Uma **função recursiva primitiva** é cada um dos membros da *menor* classe recursivamente fechada primitiva (intersecção das classes recursivamente fechadas primitivas).

---

[19]Na teoria de conjuntos ZF (Zermelo-Fraenkel).

[20]Seja f^k uma função de domínio $\mathbb{N}^k$ e $g_0^m, ..., g_{k-1}^m$ funções de domínio $\mathbb{N}^m$.
A superposição $f^k \circ \langle g_0^m, ..., g_{k-1} \rangle$ é a função $h^m(\overrightarrow{a}) = f^k\big(g_0^m(\overrightarrow{a}), ..., g_{k-1}^m(\overrightarrow{a})\big)$. No caso $k = 1$ a superposição reduz-se a uma composição de duas funções.

Resulta da definição que, f é recursiva primitiva se e só se existe uma sequência finita de funções $f_0, ..., f_n$ tal que $f_n = f$ e para cada $j \leq n$, ou $f_j \in \mathbf{F}$ por (i), (ii) ou (iii), ou então f_j é obtida por meio de superposição ou recursividade primitiva a partir de algumas das funções anteriores a f_j na sequência.

Sendo assim, se as funções f_j verificarem $f_j(\overline{y}) = \varphi(j, \overline{y})$, para alguma função φ, e $f_0 = \varphi(0, \overline{y})$ for recursiva primitiva, poderemos provar que f é recursiva primitiva utilizando o axioma de indução Ind.

3.3.7 Exemplos

1. As funções constantes são recursivas primitivas.

Fixemos o numeral $\overline{m}$ e seja $f : \mathbb{N}^k \to \mathbb{N}$ tal que $f(\overrightarrow{x}) = \overline{m}$. Se $m = 0$ então f é recursiva primitiva por definição. Se $\overline{m} = \overline{n}'$ tomamos a sequência finita de funções $f_1, ..., f_n, f$ tal que $f_1 = \mathrm{C}_0$, $f_i = f'_{i-1}$ para cada $1 < i < n$ e $f = f'_n$.

2. A adição é uma função recursiva primitiva.

Ora $+(0, x) = p_0^1(x)$ e $+(y', x) = \Big(p_0^3\big(+(y, x), y, x\big)\Big)'$. Logo o resultado segue por indução.

3. A multiplicação é uma função recursiva primitiva.

Notemos que $x \times 0 = 0$ e $x \times y' = x \times y + x$. O resultado segue por indução, uma vez que a adição é uma função recursiva.

3.3.8 Definição

Uma **função é recursiva** sse é uma função <u>recursiva primitiva</u> ou é obtida a partir de funções recursivas primitivas <u>iniciais</u> por meio de um número finito de aplicações da <u>regra</u> do operador restricto μ (**Mendelson, 1997**):

Assumindo que $g(\overrightarrow{x}, y)$ é uma função tal que, para cada $\overrightarrow{x} \in \mathbb{N}^k$, existe pelo menos um y tal que $g(\overrightarrow{x}, y) = 0$ e que $\mu y\big(g(\overrightarrow{x}, y) = 0\big)$ denota o menor y nessas condições, dizemos que a função f definida por $f(\overrightarrow{x}) = \mu y\big(g(\overrightarrow{x}, y) = 0\big)$ é obtida a partir de g por meio do operador restricto μ.

3.3.9 Observação

Não nos iremos deter mais na extensão da noção de função recursiva[21] por não ser necessária ao que se segue. Referimos, no entanto, que há diferentes formalizações

[21]Segundo a famosa Tese de Church, uma função é recursiva sse é computável por um algoritmo ou procedimento efectivo (**Enderton, 2001**), ou seja, o conceito de recursividade traduz exactamente o conceito **informal** de decidibilidade. Recordemos que um conjunto é decidível sse existe um processo de decisão para o conjunto, ou seja, um algoritmo que permita decidir, num número finito de passos, para cada entidade s, se s é ou não um elemento do conjunto.

equivalentes para esta classe de funções, como por exemplo, funções computáveis pela máquina de Turing (**Rogers, 1967**).

Vamos assumir que **AP** contém termos para todas as funções recursivas primitivas. Designemos esses termos por *termos recursivos primitivos*.

3.3.10 Definição

Uma **fórmula recursiva primitiva standard** é uma fórmula aritmética do tipo $f(\vec{x}) = 0$, onde f é um termo recursivo primitivo.

3.3.11 Definição

Uma **fórmula standard de Σ_1** é uma fórmula do tipo $\exists x\varphi(x, \vec{y})$, onde $\varphi(x, \vec{y})$ é uma fórmula recursiva primitiva standard.

3.3.12 Definição

Uma fórmula aritmética ψ **é Σ_1-demonstrável** sse é uma fórmula equivalente, na teoria **AP**, a uma fórmula standard de Σ_1. (A equivalência é demonstrável em **AP**)

3.3.13 Exemplos

Notemos que todas as fórmulas atómicas são Σ_1-demonstráveis, já que qualquer fórmula atómica é de um dos tipos $t_1 = 0$, $t_1 = t_2$, $t_1{}' = t_2$, $t_1 + t_2 = t_3$, $t_1 \times t_2 = t_3$ ou $x < y$.

Ora $t_1{}' = t_2$, por exemplo, é equivalente a $\exists x(t_1{}' = x)$.

Observemos ainda que $x < y$ é Σ_1-demonstrável uma vez que é equivalente a $\exists z(x + z' = y)$.

3.3.14 Definição

Uma fórmula aritmética ψ **é Δ_1-demonstrável** sse ψ **e** $\neg\psi$ são fórmulas equivalentes, na teoria **AP**, a fórmulas standard de Σ_1.

As classes de funções que acabámos de definir gozam de propriedades muito importantes. Na teoria **AP**, sendo S uma sentença Σ_1-demonstrável verdadeira, então é derivável (**Boolos, 1993**, pag.25)[22]. Logo, se uma sentença Δ_1-demonstrável é verdadeira, é derivável. Consequentemente, é também derivável, em **AP**, a negação de uma sentença Δ_1-demonstrável falsa. Efectivamente, sendo F e $\neg F$ Σ_1-demonstráveis, segue do primeiro

[22]George Boolos utiliza uma definição ligeiramente diferente de fórmula Σ_1-demonstrável. Uma fórmula Σ_1-demonstrável é uma fórmula equivalente, em AP, a uma fórmula de Σ_1, sendo esta um dos membros da menor classe que contém todas as fórmulas do tipo $u = v$, $u = 0$, $u' = v$, $u + v = w$ e $uv = w$ e do tipo $(F \wedge G)$, $(F \wedge G)$, $\exists xF$, $\forall x < yF$ (sempre que contenha F e G).

resultado que, aquela que for verdadeira é um teorema de **AP**. Se F é verdadeira $\vdash F$ e se F é falsa ($\neg F$ é verdadeira) $\vdash \neg F$.

3.3.15 Definição

Um **predicado prova** é uma fórmula Δ_1-demonstrável *Prf*, tal que para cada sentença aritmética φ: **AP** $\vdash \varphi$ sse para algum $n \in \mathbb{N}$, $Prf(n, \#\varphi)$[23]

$Prf(n, \#\varphi)$ traduz que "n é o código de uma derivação da sentença φ"

(Recordemos que $\#\varphi$ é o número de Gödel de φ ou código da sentença φ.)

3.3.16 Definição

Um predicado prova *Prf* diz-se **normal** sse:

1. Para cada prova k, o conjunto $T(k) = \{l : Prf(k, l)\}$ é finito. Ou seja, a função que transforma k no conjunto de numerais $T(k)$ é computável.

2. Para quaisquer números naturais k e j, existe um número natural m tal que $T(k) \cup T(j) \subseteq T(m)$. Por outras palavras, $\overline{m}$ prova todas as sentenças que $\overline{k}$ e $\overline{j}$ provam.

3.3.17 Observação

A primeira afirmação indica que cada k prova um número finito de fórmulas.

A segunda afirmação indica que um predicado prova normal é multi-conclusivo, ou seja, para cada fórmula há várias provas e não uma única.

3.3.18 Proposição

Para cada predicado prova normal *Prf*, existem funções computáveis $m(x, y)$, $a(x, y)$ e $c(x)$ tais que, quaisquer que sejam as fórmulas aritméticas φ, ψ e quaisquer que sejam os números naturais k e n, as seguintes fórmulas aritméticas são verdadeiras:

$$Prf(k, \#(\varphi \to \psi)) \wedge Prf(n, \#\varphi) \to Prf(m(k, n), \#\psi)$$
$$Prf(k, \#\varphi) \to Prf(a(k, n), \#\varphi)$$
$$Prf(n, \#\varphi) \to Prf(a(k, n), \#\varphi)$$
$$Prf(k, \#\varphi) \to Prf\Big(c(k), \#\big(Prf(k, \#\varphi)\big)\Big)$$

Demonstração

Sejam k e n tais que $\#(\varphi \to \psi) \in T(k)$ e $\#\varphi \in T(n)$, ou seja, $Prf(k, \#(\varphi \to \psi))$ e $Prf(n, \#\varphi)$ são verdadeiras. Então, por definição de predicado prova, **AP** $\vdash \varphi \to \psi$ e **AP** $\vdash \varphi$. Logo, **AP** $\vdash \psi$ por Modus Ponens. Novamente por definição de *Prf*, existe um número

[23] Por uma questão de simplificação de escrita iremos omitir as barras sobre os numerais $\overline{n}$ e $\overline{\#\varphi}$.

natural i tal que *Prf*$(i, \#\psi)$ é verdadeira. Escolhemos o menor i nessas condições, digamos j, e tomamos $m(k, n) = j$.

Sejam k e n tais que *Prf*$(k, \#\varphi)$ e *Prf*$(n, \#\varphi)$ são verdadeiras. Por definição de predicado normal, existe pelo menos um número natural i tal que $T(i) \supseteq T(k) \cup T(n)$. Escolhemos novamente o menor i que verifique a condição, digamos j, e tomamos $a(k, n) = j$.

Finalmente, seja k tal que *Prf*$(k, \#\varphi)$ é verdadeira.

Como *Prf* é Δ_1-demonstrável e *Prf*$(k, \#\varphi)$ é verdadeira em **AP**, então **AP** $\vdash$ *Prf*$(k, \#\varphi)$.

Logo, por definição de predicado prova, para cada φ tal que $\varphi \in T(k)$, existe um natural i tal que *Prf*$(i, \#$*Prf*$(k, \#\varphi))$.

Como apenas existe um número finito de sentenças nessas condições, pois $T(k)$ é finito pela primeira propriedade de normalidade de *Prf*, fixemos um i para cada uma delas, por exemplo, o menor que verifica a condição. Aplicando a segunda propriedade de normalidade de *Prf*, existe um natural j tal que *Prf*$(j, \#$*Prf*$(k, \#\varphi))$, qualquer que seja a sentença φ nas condições dadas. Tomemos então para $c(k)$ o menor dos j nessas condições. ■

3.3.19 Exemplo

A existência de um predicado prova normal é-nos garantida pelo predicado prova aritmético Prova(x, y), que traduz "x é o código de uma derivação da fórmula de código y".

De facto, cada asserção particular do tipo Prova$(n, \#F)$ é decidível, verificando-se que:

Se p é uma prova de F então $\vdash$ Prova$(\#p, \#F)$.

Se n não é o numeral correspondente ao número de Gödel de uma prova de F então $\vdash \neg$ Prova$(n, \#F)$.

Recordemos o processo de godelização (ou codificação) descrito na secção 3.2 (pág. 44). As correspondências entre cada número de Gödel e cada sentença aritmética e entre cada número de Gödel e cada derivação de AP, são injectivas. A cada sentença/derivação corresponde um único número. Dado um número natural, se for número de de uma sentença/derivação é o único nessas condições (pela unicidade da decomposição em factores primos).

Notemos que 0 considerado como símbolo tem o código 10, mas considerado como expressão tem o código 2^{10} e considerando a derivação cuja única expressão é 0, o seu código seria $3^{2^{10}}$. Ou seja, o código de uma derivação é sempre um número maior que o código da sentença que prova. Sendo assim, cada k prova um número finito de sentenças.

Por outro lado, tomemos dois números naturais k e l. Através da decomposição em factores primos de cada um deles podemos descobrir a derivação (sequência finita de fórmulas) associada a cada um. Construindo uma derivação com todas as linhas das duas derivações anteriores (eliminando possíveis repetições) e determinando o respectivo número de Gödel m dessa derivação, concluímos que $\overline{m}$ prova todas as sentenças que $\overline{k}$ e $\overline{l}$ provam.

Logo, o predicado prova aritmético Prova(x, y) é normal. No que se segue iremos utilizar a notação mais simplificada $P(x, y)$, para este predicado prova.

Finalmente reunimos as condições para definir a interpretação desejada da semântica $\mathcal{LP}$.

3.3.20 Definição

Uma interpretação aritmética para a linguagem de $\mathcal{LP}$, que denotaremos por *, inclui:

- Um predicado prova normal, *Prf*, com funções computáveis m, a e c como na proposição 3.3.18.
- Uma valuação para as letras proposicionais, por meio de sentenças aritméticas.
- Uma valuação para as variáveis prova e constantes prova, por meio de números naturais.

Além disso uma interpretação $*$ comuta com os conectivos booleanos, ou seja, $(A \wedge B)^* = (A^* \wedge B^*)$, $(A \vee B)^* = (A^* \vee B^*)$, $(A \rightarrow B)^* = (A^* \rightarrow B^*)$, comportando-se da seguinte forma com os símbolos funcionais e predicativos:

$$
\begin{aligned}
&(t.s)^* = m(t^*, s^*) \\
&(t + s)^* = a(t^*, s^*) \\
&(!t)^* = c(t^*) \\
&(t : F)^* = Prf(\overline{t^*}, \overline{\#F^*})
\end{aligned}
$$

3.3.21 Observação

1. Sob uma interpretação $*$, uma prova polinómio t é transformada num número natural t^* e uma fórmula F de $\mathcal{LP}$ é transformada numa sentença aritmética F^*.

2. A interpretação de . é a função m (associada ao predicado) que prova ψ sempre que t^* prova $\varphi \rightarrow \psi$ e s^* prova φ. A interpretação de $+$ é a função a que prova φ sempre que t^* ou s^* provam φ. Finalmente, a interpretação de ! é a função c que prova que efectivamente t^* prova a sentença φ.

3. A interpretação de $(t : F)^*$ é precisamente a que desejávamos, ou seja, t^* é uma prova de F^*. Para qualquer termo t e fórmula F, $(t : F)^*$ é Δ_1-demonstrável.

3.3.22 Notação

Seja X um conjunto de fórmulas de $\mathcal{LP}$. Então $X^* = \{F^* : F \in X\}$.

3.3.23 Definição

Uma interpretação aritmética de uma especificação de constantes EC, EC*, é uma **interpretação-EC** se todas as suas fórmulas são verdadeiras, ou seja, demonstráveis em **AP**.[24]

3.3.24 Definição

1. Uma fórmula F de $\mathcal{LP}$ é **válida** (com respeito à semântica aritmética) se F^* é verdadeira sob todas as interpretações *.

2. Uma fórmula F de $\mathcal{LP}$ é **EC-válida** se F^* é verdadeira sob todas as interpretações-EC.

3.4 Validade de $\mathcal{LP}$

3.4.1 Proposição (Validade aritmética de $\mathcal{LP}_0$)

1. Se $\mathcal{LP}_0 \vdash F$ então F é válida.

2. Se $\mathcal{LP}_0 \vdash F$ então **AP** $\vdash F^*$, qualquer que seja *.

Demonstração

Suponhamos que $\mathcal{LP}_0 \vdash F$ e demonstremos a proposição por indução na derivação em $\mathcal{LP}_0$.

- Casos em que F é um dos axiomas A_0 - A_4.

A_0. Suponhamos que F é uma axioma-esquema da lógica proposicional clássica.

Neste caso, é imediato que a sentença aritmética correspondente F^* é verdadeira e derivável em **AP**.

A_1. Suponhamos que F é do tipo $t : A \to A$

Seja * uma interpretação qualquer. Então $(t : A \to A)^* \equiv Prf(\overline{t^*}, \overline{\#A^*}) \to A^*$.

1. Provemos que $(t : A \to A)^*$ é verdadeira.

Se $Prf(\overline{t^*}, \overline{\#A^*})$ é verdadeira então t^* é uma prova de A^* e logo A^* é verdadeira. Logo a implicação $Prf(\overline{t^*}, \overline{\#A^*}) \to A^*$ é verdadeira.

2. Provemos que **AP** $\vdash (t : A \to A)^*$.

Se $Prf(\overline{t^*}, \overline{\#A^*})$ é verdadeira então, por definição de predicado prova, **AP** $\vdash A^*$. Logo, **AP** $\vdash Prf(\overline{t^*}, \overline{\#A^*}) \to A^*$, ou seja, **AP** $\vdash (t : A \to A)^*$.

Se $Prf(\overline{t^*}, \overline{\#A^*})$ é falsa então, sendo Prf uma fórmula Δ_1-demonstrável, **AP** $\vdash \neg Prf(\overline{t^*}, \overline{\#A^*})$. Logo, **AP** $\vdash Prf(\overline{t^*}, \overline{\#A^*}) \to A^*$, ou seja, **AP** $\vdash (t : A \to A)^*$.[25]

---

[24]Todas as fórmulas de EC* são do tipo $Prf(c^*, \#A^*)$, onde c é uma prova constante e A um axioma do tipo $A_0 - A_4$. Como Prf é Δ_1-demonstrável, se $Prf(c^*, \#A^*)$ é verdadeira, então é demonstrável em **AP**.

[25]Ao longo desta demonstração utilizamos frequentemente os seguintes resultados:
Se AP $\vdash \varphi$ então AP $\vdash \psi \to \varphi$ e se AP $\vdash \neg\varphi$ então AP $\vdash \varphi \to \psi$.

$\mathbf{A_2}$. Suponhamos que F é do tipo $t:(A\to B)\to(s:A\to(ts):B)$

Seja * uma interpretação qualquer. Então:

$\left(t:(A\to B)\to(s:A\to(ts):B)\right)^*\equiv Prf(t^*,\#(A\to B)^*)\to\left(Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)\right)$

1. Sejam k e n números naturais tais que $k=t^*$ e $n=s^*$ e φ, ψ sentenças aritméticas tais que $\varphi=A^*$ e $\psi=B^*$. Pela Proposição 3.3.18, existe uma função computável m tal que:

$Prf(k,\#(\varphi\to\psi))\wedge Prf(n,\#\varphi)\to Prf(m(k,n),\#\psi)$ é verdadeira.

Logo, $Prf(k,\#(\varphi\to\psi))\to\left(Prf(n,\#\varphi)\to Prf(m(k,n),\#\psi\right)$ também é verdadeira e sendo $(t.s)^*=m(k,n)$ pela definição 3.3.20, concluimos que F^* é verdadeira.

2. Provemos que **AP** $\vdash Prf(t^*,\#(A\to B)^*)\to\left(Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)\right)$

Por 1., $Prf(t^*,\#(A\to B)^*)\to\left(Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)\right)$ é verdadeira.

Suponhamos que $Prf(t^*,\#(A\to B)^*)$ é falsa. Então **AP** $\vdash\neg Prf(t^*,\#(A\to B)^*)$ e logo **AP** $\vdash Prf(t^*,\#(A\to B)^*)\to\left(Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)\right)$.

Suponhamos agora que $Prf(t^*,\#(A\to B)^*)$ é verdadeira.

Então, por 1, $\left(Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)\right)$ também é verdadeira.

Se $Prf(s^*,\#A^*)$ for verdadeira então $Prf((ts)^*,\#B^*)$ é verdadeira e sendo Prf Δ_1-demonstrável, **AP** $\vdash Prf((ts)^*,\#B^*)$. Logo, **AP** $\vdash Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)$ e finalmente, **AP** $\vdash Prf(t^*,\#(A\to B)^*)\to\left(Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)\right)$.

Se $Prf(s^*,\#A^*)$ for falsa, **AP** $\vdash\neg Prf(s^*,\#A^*)$, **AP** $\vdash Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)$ e novamente **AP** $\vdash Prf(t^*,\#(A\to B)^*)\to\left(Prf(s^*,\#A^*)\to Prf((ts)^*,\#B^*)\right)$.

$\mathbf{A_3}$. Suponhamos que F é do tipo $(t:A)\to(!t:(t:A))$

Seja * uma interpretação qualquer. Então:

$(t:A\to !t:(t:A))^*\equiv Prf(t^*,\#A^*)\to Prf\left((!t)^*,\#Prf(t^*,\#A^*)\right)$

1. Sejam k um número natural tal que $k=t^*$ e φ uma sentença aritmética tal que $\varphi=A^*$. Pela Proposição 3.3.18, existe uma função computável c tal que:

$Prf(k,\#\varphi)\ \to\ Prf\left(c(k),\#(Prf(k,\#\varphi))\right)$ é verdadeira. Como, pela definição 3.3.20, $(!t)^*=c(k)$, concluimos que F^* é verdadeira.

2. Provemos que **AP** $\vdash Prf(t^*,\#A^*)\to Prf\left((!t)^*,\#Prf(t^*,\#A^*)\right)$.

Suponhamos que $Prf(t^*,\#A^*)$ é falsa. Então **AP** $\vdash\neg Prf(t^*,\#A^*)$ e logo **AP** $\vdash Prf(t^*,\#A^*)\to Prf\left((!t)^*,\#Prf(t^*,\#A^*)\right)$.

Suponhamos que $Prf(t^*,\#A^*)$ é verdadeira. Por 1, $Prf\left((!t)^*,\#Prf(t^*,\#A^*)\right)$ é verdadeira.

Logo **AP** $\vdash Prf\left((!t)^*,\#Prf(t^*,\#A^*)\right)$ e portanto **AP** $\vdash\left(t:A\to !t:(t:A)\right)^*$.

$\mathbf{A_4}$. Suponhamos que F é do tipo $t : A \rightarrow (t + s) : A$

Seja * uma interpretação qualquer. Então :

$(t : A \rightarrow (s + t) : A)^* \equiv Prf(t^*, \#A^*) \rightarrow Prf\left((s{+}t)^*, \#A^*\right)$

1. Sejam k e n números naturais tais que $k = t^*$, $n = s^*$ e seja φ uma sentença aritmética tal que $\varphi = A^*$. Pela Proposição 3.3.18, existe uma função computável a tal que $Prf(k, \#\varphi) \rightarrow Prf(a(k, n), \#\varphi)$ é verdadeira. Sendo $(t + s)^* = a(t^*, s^*)$, pela definição 3.3.20, podemos concluir que F^* é verdadeira.

2. Provemos que **AP** $\vdash Prf(t^*, \#A^*) \rightarrow Prf\left((s{+}t)^*, \#A^*\right)$.

Suponhamos que $Prf(t^*, \#A^*)$ é falsa. Então **AP** $\vdash \neg Prf(t^*, \#A^*)$ e logo **AP** $\vdash Prf(t^*, \#A^*) \rightarrow Prf\left((s{+}t)^*, \#A^*\right)$.

No caso de $Prf(t^*, \#A^*)$ ser verdadeira, por 1, $Prf\left((s{+}t)^*, \#A^*\right)$ também é verdadeira e logo **AP** $\vdash Prf\left((s{+}t)^*, \#A^*\right)$. Portanto, **AP** $\vdash (t : A \rightarrow (s + t) : A)^*$.

- Caso em que F é introduzido, na derivação, por meio da regra R_1.

$\mathbf{R_1}$. Suponhamos que $\mathcal{LP}_0 \vdash F$ e F é introduzida por meio de R_1, a partir de $G \rightarrow F$ e de G. Por hipótese de indução:

1. $G \rightarrow F$ é G são válidas.
2. **AP** $\vdash (G \rightarrow F)^*$ e **AP** $\vdash G^*$, qualquer que seja *.

Seja * uma interpretação qualquer.

1. Por hipótese de indução, $G^* \rightarrow F^*$ e G^* são verdadeiras. Logo F^* é verdadeira.

2. Sejam φ e ψ sentenças aritméticas tais que $\varphi = G^*$ e $\psi = F^*$. Por hipótese de indução, **AP** $\vdash \varphi \rightarrow \psi$ e **AP** $\vdash \varphi$. Logo, por Modus Ponens, **AP** $\vdash \psi$. ■

3.4.2 Corolário (Validade aritmética de $\mathcal{LP}$)

1. Se $\mathcal{LP}$(EC) $\vdash F$ então F é EC-válida.
2. Se $\mathcal{LP}$(EC) $\vdash F$ então **AP** $\vdash F^*$, qualquer que seja a interpretação-EC.

Demonstração

Recordemos que $\mathcal{LP}$(EC) não é mais do que $\mathcal{LP}_0$ com a especificação de constantes EC.

Suponhamos então que $\mathcal{LP}$(EC) $\vdash F$.

Se a derivação se reduz a uma derivação em $\mathcal{LP}_0$, então, pela proposição anterior, F é válida e **AP** $\vdash F^*$.

Se F é introduzido, na derivação, por meio da regra R_2, então é uma fórmula do tipo $c : A$, pertencente a EC, em que c é uma constante prova e A é um dos axiomas A_0-A_4. Nesse caso F é verdadeira sob todas as interpretações EC, por definição de interpretação-EC.

Por outro lado, **AP** $\vdash F^*$, qualquer que seja a interpretação-EC, já que **AP** $\vdash Prf(c^*, \#A^*)$ (novamente por definição de interpretação-EC e por *Prf* ser Δ_1-demonstrável). ■

3.5 Completude aritmética de $\mathcal{LP}$

Nesta secção iremos utilizar uma formulação de $\mathcal{LP}$ e $\mathcal{LP}_0$ por meio de sequências. Notaremos estes sistemas por $\mathcal{LPG}$ e $\mathcal{LPG}_0$, respectivamente.

3.5.1 Definição

Designamos por sequência (ou sequente), um par ordenado de conjuntos finitos de fórmulas de $\mathcal{LP}$, (Γ, Δ), em que Γ é o antecedente e Δ o sucedente. A notação que utilizamos para a sequência é $\Gamma \Rightarrow \Delta$[26].

Antes de apresentarmos os sistemas $\mathcal{LPG}$ e $\mathcal{LPG}_0$, façamos um esboço do trabalho que iremos desenvolver nesta secção com o objectivo de obter a completude aritmética de $\mathcal{LP}$.

Após a apresentação do cálculo de sequentes correspondente a $\mathcal{LP}$ e $\mathcal{LP}_0$, começaremos por demonstrar que uma sequência $\Gamma \Rightarrow \Delta$ é derivável se e só se a fórmula $\bigwedge\Gamma \rightarrow \bigvee\Delta$ é derivável (Proposição 3.5.7).

Em seguida, descrevemos um processo para obter, a partir de uma qualquer sequência não derivável, uma sequência saturada não derivável (Lema 3.5.10). Apresentamos ainda um algoritmo para completar uma tal sequência saturada com todas as fórmulas que é possível obter aplicando as operações $!$, $.$ e $+$ a todas as fórmulas já existentes (Lema 3.5.12). Estas três operações básicas (verificador de provas, aplicação e escolha) incorporadas em $\mathcal{LP}$, constituem a base para todas as operações com provas exprimíveis na linguagem proposicional adoptada.

O passo seguinte será definir uma interpretação aritmética das fórmulas de $\mathcal{LP}$ (na demonstração do Teorema 3.5.13) e estabelecer a completude de $\mathcal{LP}$ através de um raciocínio por absurdo. Na semântica associada à linguagem de $\mathcal{LP}$ verifica-se que a interpretação de "t é uma prova de F" é uma fórmula aritmética que traduz uma afirmação sobre os códigos de t e F.

No final da secção (Corolário 3.5.22) poderemos concluir que, se uma fórmula aritmética for derivável na teoria **AP** (Aritmética de Peano) e for uma interpretação de uma fórmula F de $\mathcal{LP}$, então F é uma fórmula derivável em $\mathcal{LP}$. Reciprocamente, qualquer que seja a fórmula derivável em $\mathcal{LP}$, a sua interpretação aritmética é derivável em **AP**.

[26]Intuitivamente, uma sequência $A_1, A_2, \ldots A_n \Rightarrow B_1, B_2, \ldots B_n$ significa que a disjunção $B_1 \vee B_2 \vee \ldots \vee B_n$ resulta da conjunção $A_1 \wedge A_2 \wedge A_n$. A utilização de sequências é conveniente para sabermos, em cada passo da derivação, quais as hipóteses que estão activas.

3.5.2 Definição

Sistemas $\mathcal{LPG}$ e $\mathcal{LPG}_0$

- Axiomas de $\mathcal{LPG}_0$

  Sequências do tipo: $\Gamma,\ F \Rightarrow F, \Delta$ ou $\Gamma,\ \perp\ \Rightarrow \Delta$

- Regras de $\mathcal{LPG}_0$

  Regras de Gentzen usuais na lógica proposicional clássica (**Troelstra e Schwichtenberg, 2000**):

Regra do corte: $\dfrac{\Gamma_1 \Rightarrow \Delta_1, A \quad A, \Gamma_2 \Rightarrow \Delta_2}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2}$

Regras de introdução dos operadores lógicos:

Utilizamos $L\diamond$ e $R\diamond$, para indicar a regra de introdução da fórmula com o operador $\diamond$, à esquerda e à direita, respectivamente.

$$\frac{A,\Gamma\Rightarrow\Delta}{A\wedge B,\Gamma\Rightarrow\Delta} \text{ ou } \frac{B,\Gamma\Rightarrow\Delta}{A\wedge B,\Gamma\Rightarrow\Delta}\ (L\wedge) \qquad \frac{\Gamma\Rightarrow\Delta,A \quad \Gamma\Rightarrow\Delta,B}{\Gamma\Rightarrow\Delta,A\wedge B}\ (R\wedge)$$

$$\frac{A,\Gamma\Rightarrow\Delta \quad B,\Gamma\Rightarrow\Delta}{A\vee B,\Gamma\Rightarrow\Delta}\ (L\vee) \qquad \frac{\Gamma\Rightarrow\Delta,A}{\Gamma\Rightarrow\Delta,A\vee B} \text{ ou } \frac{\Gamma\Rightarrow\Delta,B}{\Gamma\Rightarrow\Delta,A\vee B}\ (R\vee)$$

$$\frac{\Gamma\Rightarrow\Delta,A \quad B,\Gamma\Rightarrow\Delta}{A\to B,\Gamma\Rightarrow\Delta}\ (L\to) \qquad \frac{A,\Gamma\Rightarrow\Delta,B}{\Gamma\Rightarrow\Delta,A\to B}\ (R\to)$$

Regras estruturais de contracção:

$$\frac{A,A,\Gamma\Rightarrow\Delta}{A,\Gamma\Rightarrow\Delta}\ (LC) \qquad \frac{\Gamma\Rightarrow\Delta,A,A}{\Gamma\Rightarrow\Delta,A}\ (RC)$$

E ainda as seguintes regras:

$$\frac{A,\Gamma\Rightarrow\Delta}{t{:}A,\Gamma\Rightarrow\Delta}\ (:\ \Rightarrow) \qquad \frac{\Gamma\Rightarrow\Delta,t{:}A}{\Gamma\Rightarrow\Delta,!t{:}t{:}A}\ (\Rightarrow !)$$

$$\frac{\Gamma\Rightarrow\Delta,t{:}A}{\Gamma\Rightarrow\Delta,(t+s){:}A}\ (\Rightarrow\ +) \qquad \frac{\Gamma\Rightarrow\Delta,t{:}A}{\Gamma\Rightarrow\Delta,(s+t){:}A}\ (\Rightarrow\ +)$$

$$\frac{\Gamma\Rightarrow\Delta,t{:}(A\to B) \quad \Gamma\Rightarrow\Delta,s{:}A}{\Gamma\Rightarrow\Delta,(ts){:}B}\ (\Rightarrow\ .)$$

Para obter o sistema $\mathcal{LPG}$, acrescentamos ao sistema $\mathcal{LPG}_0$ a regra $\dfrac{}{\Gamma\Rightarrow c{:}A,\Delta}\ (\Rightarrow c)$.

3.5.3 Observações

1. Em cada uma das regras anteriores, Γ e Δ constituem o contexto de aplicação da regra. Na conclusão, a fórmula introduzida (na qual consta o operador $\diamond$) diz-se a fórmula principal. As fórmulas da(s) permissa(s) das quais a fórmula principal é concluída (e que não pertencem ao contexto) são as fórmulas activas.

2. Uma outra versão da regra do corte, aparentemente mais simples, mas que obriga a que o contexto utilizado seja o mesmo, é o seguinte caso particular ($\Gamma_1 = \Gamma_2$, $\Delta_1 = \emptyset$, $\Delta_2 = \{B\}$):

$$\frac{\Gamma \Rightarrow A \quad A, \Gamma \Rightarrow B}{\Gamma \Rightarrow B}$$

Observemos ainda que a regra do corte traduz uma espécie de transitividade que nos vai permitir, nomeadamente, verificar a correspondência entre as regras de introdução dos conectivos à esquerda e as regras de eliminação dos mesmos conectivos.

3.5.4 Notação

Retirando a regra do corte a $\mathcal{LPG}$ e $\mathcal{LPG}_0$, obtemos os sistemas $\mathcal{LPG}^-$ e $\mathcal{LPG}_0^-$, respectivamente.

Em seguida, e a título de exemplo, apresentamos algumas deduções em $\mathcal{LPG}$.

Derivamos, nomeadamente, regras correspondentes à eliminação dos conectivos e apresentamos um exemplo de derivação sem hipóteses, válida portanto em qualquer contexto.

3.5.5 Lema

1. $\Gamma \Rightarrow A \wedge B \vdash \Gamma \Rightarrow A$

2. $\Gamma \Rightarrow A \rightarrow B \quad \Delta \Rightarrow A \vdash \Gamma, \Delta \Rightarrow B$

3. $\Gamma \Rightarrow A \vee B \quad A \Rightarrow C \quad B \Rightarrow C \vdash \Gamma \Rightarrow C$

4. $\vdash \ \Rightarrow (A \rightarrow B) \vee (B \rightarrow A)$

Demonstração

1. $\Gamma \Rightarrow A \wedge B \vdash \Gamma \Rightarrow A$

Utilizando a versão geral da regra do corte:

$$\frac{\Gamma \Rightarrow A \wedge B \quad \dfrac{A \Rightarrow A}{A \wedge B \Rightarrow A}}{\Gamma \Rightarrow A}$$

Poderíamos utilizar sempre o mesmo contexto, tirando benefício da nossa opção em adoptar axiomas generalizados. Verificamos, no entanto, que esta derivação apresenta uma leitura ligeiramente mais complexa:

$$\dfrac{\Gamma \Rightarrow A \wedge B \quad \dfrac{A, \Gamma \Rightarrow A}{A \wedge B, \Gamma \Rightarrow A}}{\Gamma \Rightarrow A}$$

2. $\Gamma \Rightarrow A \rightarrow B \quad \Delta \Rightarrow A \vdash \Gamma, \Delta \Rightarrow B$

$$\dfrac{\Gamma \Rightarrow A \rightarrow B \quad \dfrac{\Delta \Rightarrow A \quad B \Rightarrow B}{\Delta, A \rightarrow B \Rightarrow B}}{\Gamma, \Delta \Rightarrow B}$$

3. $\Gamma \Rightarrow A \vee B \quad A \Rightarrow C \quad B \Rightarrow C \vdash \Gamma \Rightarrow C$

$$\dfrac{\Gamma \Rightarrow A \vee B \quad \dfrac{A \Rightarrow C \quad B \Rightarrow C}{A \vee B \Rightarrow C}}{\Gamma \Rightarrow C}$$

4. $\vdash \ \Rightarrow (A \rightarrow B) \vee (B \rightarrow A)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A, B \Rightarrow A, B}{A \Rightarrow B, B \rightarrow A}}{\Rightarrow A \rightarrow B, B \rightarrow A}}{\Rightarrow A \rightarrow B, (A \rightarrow B) \vee (B \rightarrow A)}}{\Rightarrow (A \rightarrow B) \vee (B \rightarrow A), (A \rightarrow B) \vee (B \rightarrow A)}}{\Rightarrow (A \rightarrow B) \vee (B \rightarrow A)}$$

■

3.5.6 Observação

Podemos encarar cada dedução como uma *árvore* que parte da(s) hipótese(s) (*raíz*) e, por meio de introdução de axiomas e aplicação das regras (*folhas*), vai construindo outros sequentes até obter a sequência conclusão ou tese da dedução.

3.5.7 Proposição

$\mathcal{LPG}_0 \vdash \Gamma \Rightarrow \Delta$ se e só se $\mathcal{LP}_0 \vdash \bigwedge\Gamma \rightarrow \bigvee\Delta$.

$\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$ se e só se $\mathcal{LP} \vdash \bigwedge\Gamma \rightarrow \bigvee\Delta$.

Demonstração

Demonstramos as duas implicações da afirmação $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$ se e só se $\mathcal{LP} \vdash \bigwedge\Gamma \to \bigvee\Delta$, por indução na derivação em $\mathcal{LPG}$ ou na complexidade de $\Gamma \Rightarrow \Delta$.

Sem perda de generalidade, utilizamos versões simplificadas de $L\diamond$ (com $\Gamma = \emptyset$) e de $R\diamond$ (com $\Delta = \emptyset$), para simplicidade da demonstração.

Pelo Teorema da Dedução (3.2.12), $\mathcal{LP} \vdash \bigwedge\Gamma \to \bigvee\Delta$ se e só se $\bigwedge\Gamma \vdash \bigvee\Delta$ em $\mathcal{LP}$.

1. Se $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$ então $\bigwedge\Gamma \vdash \bigvee\Delta$ em $\mathcal{LP}$.

- Casos básicos

Se $\Gamma \Rightarrow \Delta$ é um dos axiomas de $\mathcal{LPG}$:

Se $\mathcal{LPG} \vdash \Gamma, F \Rightarrow F$ então $\bigwedge\Gamma \wedge F \vdash F$ em $\mathcal{LP}$ (as duas derivações são imediatas)

Se $\mathcal{LPG} \vdash \Gamma, \bot \Rightarrow \Delta$ então $\bigwedge\Gamma \wedge \bot \vdash \bigvee\Delta$ em $\mathcal{LP}$ (já que $\bigwedge\Gamma \wedge \bot \vdash \bot$ e $\bot \vdash \bigvee\Delta$ por lógica proposicional)

- Passo de indução

Se $\Gamma \Rightarrow \Delta$ é introduzida por meio da regra:

$(: \Rightarrow)$ Neste caso $\Gamma \Rightarrow \Delta$ é do tipo $t : A, \Theta \Rightarrow \Delta$, introduzida a partir da hipótese $A, \Theta \Rightarrow \Delta$. Por hipótese de indução, se $\mathcal{LPG} \vdash A, \Theta \Rightarrow \Delta$ então $\bigwedge\Theta \wedge A \vdash \bigvee\Delta$ em $\mathcal{LP}$.

Provemos que $\bigwedge\Theta \wedge t : A \vdash \bigvee\Delta$:

1	$\bigwedge\Theta \wedge (t : A)$	H
2	$\bigwedge\Theta \wedge (t : A) \to \bigwedge\Theta$	(A_0)
3	$\bigwedge\Theta \wedge (t : A) \to t : A$	(A_0)
4	$\bigwedge\Theta$	1, 2 (MP)
5	$t : A$	1, 3 (MP)
6	$(t : A) \to A$	(A_1)
7	A	5, 6 (MP)
8	$\bigwedge\Theta \wedge A \to \bigvee\Delta$	(hip. indução)
9	$A \to (\bigwedge\Theta \to (\bigwedge\Theta \wedge A))$	(A_0)
10	$\bigwedge\Theta \to (\bigwedge\Theta \wedge A)$	7, 9 (MP)
11	$\bigwedge\Theta \wedge A$	4, 10 (MP)
12	$\bigvee\Delta$	8, 11 (MP)

$(\Rightarrow !)$ Neste caso $\Gamma \Rightarrow \Delta$ é do tipo $\Gamma \Rightarrow \Theta, !t : t : A$, introduzida a partir da hipótese $\Gamma \Rightarrow \Theta, t : A$. Por hipótese de indução, se $\mathcal{LPG} \vdash \Gamma \Rightarrow \Theta, t : A$ então $\bigwedge\Gamma \vdash \bigvee\Theta \vee (t : A)$ em $\mathcal{LP}$.

Provemos que $\bigwedge\Gamma \vdash \bigvee\Theta \vee (!t : t : A)$:

1	$\bigwedge\Gamma$	H
2	$\bigvee\Theta \vee (t : A)$	(hip. indução)
3	$(t : A) \to (!t : t : A)$	(A_3)
4	$\bigvee\Theta \vee (!t : t : A)$	2, 3 (MP e lógica proposicional)

($\Rightarrow +$) Neste caso $\Gamma \Rightarrow \Delta$ é do tipo $\Gamma \Rightarrow \Theta, (t+s) : A$, introduzida a partir da hipótese $\Gamma \Rightarrow \Theta, t : A$. Por hipótese de indução, $\bigwedge\Gamma \vdash \bigvee\Theta \vee (t : A)$ em $\mathcal{LP}$.

Provemos que $\bigwedge\Gamma \vdash \bigvee\Theta \vee (t+s) : A$:

1	$\bigwedge\Gamma$	H
2	$\bigvee\Theta \vee (t : A)$	(hip. indução)
3	$(t : A) \rightarrow (t+s) : A$	(A_4)
4	$\bigvee\Theta \vee (t+s) : A$	2, 3 (MP e lógica proposicional)

($\Rightarrow .$) Neste caso $\Gamma \Rightarrow \Delta$ é do tipo $\Gamma \Rightarrow \Theta, (ts) : B$, introduzida a partir das hipóteses $\Gamma \Rightarrow \Theta, t : (A \rightarrow B)$ e $\Gamma \Rightarrow \Theta, s : A$. Por hipótese de indução, $\bigwedge\Gamma \vdash \bigvee\Theta \vee t : (A \rightarrow B)$ e $\bigwedge\Gamma \vdash \bigvee\Theta \vee (s : A)$.

Provemos que $\bigwedge\Gamma \vdash \bigvee\Theta \vee (ts) : B$:

1	$\bigwedge\Gamma$	H
2	$\bigvee\Theta \vee t : (A \rightarrow B)$	(hip. indução)
3	$\bigvee\Theta \vee (s : A)$	(hip. indução)
4	$t : (A \rightarrow B) \rightarrow (s : A \rightarrow (ts) : B)$	(A_2)
5	$\bigvee\Theta \vee (s : A \rightarrow (ts) : B)$	2, 4 (MP e lógica proposicional)
6	$\bigvee\Theta \vee (ts : B)$	3, 5 (MP e lógica proposicional)

($\Rightarrow c$) $\Gamma \Rightarrow \Delta$ é do tipo $\Gamma \Rightarrow \Theta, c : A$.

Provemos que $\bigwedge\Gamma \vdash \bigvee\Theta \vee c : A$:

1	$\bigwedge\Gamma$	H
2	$c : A$	1 (R_2)
3	$(c : A) \rightarrow \bigvee\Theta \vee (c : A)$	2 (A_0)
4	$\bigvee\Theta \vee (c : A)$	2, 3 (MP)

(R $\wedge$) $\Gamma \Rightarrow \Delta$ é do tipo $\Gamma \Rightarrow A \wedge B$, introduzida a partir das hipóteses $\Gamma \Rightarrow A$ e $\Gamma \Rightarrow B$. Por hipótese de indução, $\bigwedge\Gamma \vdash A$ e $\bigwedge\Gamma \vdash B$.

Provemos que $\bigwedge\Gamma \vdash A \wedge B$:

1	$\bigwedge\Gamma$	H
2	A	(hip. indução)
3	B	(hip. indução)
4	$A \rightarrow (B \rightarrow A \wedge B)$	(A_0)
5	$B \rightarrow A \wedge B$	2, 4 (MP)
6	$A \wedge B$	3, 5 (MP)

(L $\wedge$) $\Gamma \Rightarrow \Delta$ é do tipo $A \wedge B \Rightarrow \Delta$, introduzida a partir da hipótese $A \Rightarrow \Delta$. Por hipótese de indução, $A \vdash \bigvee\Delta$.

Provemos que $A \wedge B \vdash \bigvee\Delta$:

1	$A \wedge B$	H
2	$A \wedge B \to A$	(A_0)
3	A	1,2 (MP)
4	$\bigvee\Delta$	(hip. indução)

($R\vee$) $\Gamma \Rightarrow \Delta$ é do tipo $\Gamma \Rightarrow A \vee B$, introduzida a partir da hipótese $\Gamma \Rightarrow A$. Por hipótese de indução, $\bigwedge\Gamma \vdash A$.

Provemos que $\bigwedge\Gamma \vdash A \vee B$:

1	$\bigwedge\Gamma$	H
2	A	(hip. indução)
3	$A \to A \vee B$	(A_0)
4	$A \vee B$	2, 3 (MP)

($L\vee$) $A \vee B \Rightarrow \Delta$ é introduzida a partir das hipóteses $A \Rightarrow \Delta$ e $B \Rightarrow \Delta$. Por hipótese de indução, $\vdash A \to \bigvee\Delta$ e $\vdash B \to \bigvee\Delta$.

Provemos que $A \vee B \vdash \bigvee\Delta$:

1	$A \vee B$	H
2	$A \to \bigvee\Delta$	(hip. indução)
3	$B \to \bigvee\Delta$	(hip. indução)
4	$A \vee B \to \big((A \to \bigvee\Delta) \to ((B \to \bigvee\Delta) \to \bigvee\Delta)\big)$	(A_0)
5	$(A \to \bigvee\Delta) \to ((B \to \bigvee\Delta) \to \bigvee\Delta)$	1, 4 (MP)
6	$(B \to \bigvee\Delta) \to \bigvee\Delta$	2, 5 (MP)
7	$\bigvee\Delta$	3, 6 (MP)

($R\to$) $\Gamma \Rightarrow A \to B$, introduzida a partir da hipótese $A, \Gamma \Rightarrow B$. Por hipótese de indução, $\bigwedge\Gamma \wedge A \vdash B$. Logo, $\bigwedge\Gamma, A \vdash B$ (por lógica proposicional) e $\bigwedge\Gamma \vdash A \to B$ pelo teorema da dedução.

($L\to$) $A \to B \Rightarrow \Delta$ é introduzida a partir das hipóteses $\Rightarrow \Delta, A$ e $B \Rightarrow \Delta$. Por hipótese de indução, $\vdash \bigvee\Delta \vee A$ e $\vdash B \to \bigvee\Delta$.

Provemos que $A \to B \vdash \bigvee\Delta$:

1	$A \to B$	H
2	$\bigvee\Delta \vee A$	(hip. indução)
3	$B \to \bigvee\Delta$	(hip. indução)
4	$\bigvee\Delta \vee A \to \big((A \to B) \to ((B \to \bigvee\Delta) \to \bigvee\Delta)\big)$	(A_0)
5	$(A \to B) \to ((B \to \bigvee\Delta) \to \bigvee\Delta)$	2, 4 (MP)
6	$(B \to \bigvee\Delta) \to \bigvee\Delta$	1, 5 (MP)
7	$\bigvee\Delta$	3, 6 (MP)

(RC) $\Gamma \Rightarrow A$ é introduzida a partir da hipótese $\Gamma \Rightarrow A, A$. Por hipótese de indução, $\Gamma \vdash A, A$. É imediato que $\Gamma \vdash A$.

(LC) $A, \Gamma \Rightarrow \Delta$ é introduzida a partir da hipótese $A, A, \Gamma \Rightarrow \Delta$. Por hipótese de indução, $A, A, \Gamma \vdash \Delta$. É imediato que $A, \Gamma \vdash \Delta$.

(Corte) $\Gamma \Rightarrow B$ é introduzida a partir das hipóteses $\Gamma \Rightarrow A$ e $A, \Gamma \Rightarrow B$. Por hipótese de indução, $\Gamma \vdash A$ e $A, \Gamma \vdash B$.

Provemos que $\Gamma \vdash B$:

1	Γ	H
2	A	(hip. indução)
3	B	1, 2 (hip. indução)

2. Se, $\mathcal{LP} \vdash \bigwedge\Gamma \rightarrow \bigvee\Delta$ então $\mathcal{LPG} \vdash \Gamma \Rightarrow \Delta$.

Na demonstração de 1. está já implícita a existência de um paralelismo entre as regras de $\mathcal{LP}$ e as regras de $\mathcal{LPG}$. Sendo assim, na demonstração desta implicação, limitamo-nos a provar os casos em que esse paralelismo não nos pareça imediato.

- Casos básicos

Se $\bigwedge\Gamma \rightarrow \bigvee\Delta$ é um dos axiomas de $\mathcal{LP}$, digamos F:

Se F é do tipo A_0 e $\mathcal{LP} \vdash F$ então $\mathcal{LPG} \vdash \ \Rightarrow F$.

Se F é do tipo A_1, digamos, $t : A \rightarrow A$ então $\mathcal{LPG} \vdash t : A \Rightarrow A$ por aplicação directa da regra ($: \ \Rightarrow$) $\frac{A \Rightarrow A}{t : A \Rightarrow A}$.

Se F é do tipo A_2, digamos, $t : (A \rightarrow B) \rightarrow (s : A \rightarrow ts : B)$ então $\mathcal{LPG} \vdash t : (A \rightarrow B) \Rightarrow (s : A \rightarrow ts : B)$ por aplicação da regra ($\Rightarrow .$) a dois axiomas, seguida de aplicação de (R$\rightarrow$):

$$\cfrac{\cfrac{t : (A \rightarrow B), s : A \Rightarrow t : (A \rightarrow B) \quad t : (A \rightarrow B), s : A \Rightarrow s : A}{t : (A \rightarrow B), s : A \Rightarrow ts : B}}{t : (A \rightarrow B) \Rightarrow (s : A \rightarrow ts : B)}$$

Se F é do tipo A_3, digamos, $t : A \rightarrow !t : (t : A)$ então $\mathcal{LPG} \vdash t : A \Rightarrow (!t : (t : A))$ por aplicação directa da regra ($\Rightarrow !$) ao axioma $t : A \Rightarrow t : A$.

Se F é do tipo A_4, digamos, $t : A \rightarrow (s + t) : A$ então $\mathcal{LPG} \vdash t : A \Rightarrow (s + t) : A$ por aplicação directa da regra ($\Rightarrow +$) ao axioma $t : A \Rightarrow t : A$.

- Passo de indução

Se $\bigwedge\Gamma \rightarrow \bigvee\Delta$ é uma fórmula de $\mathcal{LP}$, digamos F, introduzida por meio da regra:

R_1) Suponhamos que F é introduzida a partir das fórmulas $G \rightarrow F$ e G. Por hipótese de indução, $\mathcal{LPG} \vdash G \Rightarrow F$ e $\mathcal{LPG} \vdash \ \Rightarrow G$. Logo, por aplicação imediata da regra do corte, $\mathcal{LPG} \vdash \ \Rightarrow F$ ($\frac{\Rightarrow G \quad G \Rightarrow F}{\Rightarrow F}$).

R$_2$) Suponhamos que F é do tipo $c : A$ introduzida pela necessidade de axioma. Por aplicação imediata de ($\Rightarrow c$) $\mathcal{LPG} \vdash\ \Rightarrow c : A$. ■

3.5.8 Corolário

$\mathcal{LP}$(EC) $\vdash F$ sse $\mathcal{LPG}_0 \vdash$ EC $\Rightarrow F$.

Demonstração

Pela Proposição 1, fazendo $\Gamma =$ EC e $\Delta = \{F\}$, obtemos $\mathcal{LPG}_0 \vdash$ EC $\Rightarrow F$ sse $\mathcal{LP}_0 \vdash \bigwedge$EC$\rightarrow F$. Mas, por definição de $\mathcal{LP}$(EC) e pelo Teorema da Dedução (Teorema 1 da secção *3.2*), $\mathcal{LP}_0 \vdash \bigwedge$EC$\rightarrow F$ sse $\mathcal{LP}$(EC) $\vdash F$. ■

A definição que se segue é parcialmente motivada pelo processo que normalmente utilizamos na procura de uma dedução. O procedimento usual é procurar um meio para inferir determinada fórmula, observando as regras do sistema, de baixo para cima, por forma a determinar o que é necessário ter na dedução para obter a fórmula desejada.

A título de exemplo, suponhamos que se pretende que $A \rightarrow B$ pertença ao sucedente de determinado sequente. Observando (R$\rightarrow$) será então conveniente que A esteja no antecedente e B no sucedente.

3.5.9 Definição

A sequência $\Gamma \Rightarrow \Delta$ é **saturada** sse:

1. Se $A \rightarrow B \in \Gamma$ então $B \in \Gamma$ ou $A \in \Delta$.
2. Se $A \rightarrow B \in \Delta$ então $A \in \Gamma$ e $B \in \Delta$.[27]
3. Se $t : A \in \Gamma$ então $A \in \Gamma$.
4. Se $!t : t : A \in \Delta$ então $t : A \in \Delta$.
5. Se $(s + t) : A \in \Delta$ então $s : A \in \Delta$ e $t : A \in \Delta$.
6. Se $st : B \in \Delta$ então, para cada $X \rightarrow B$ que ocorre como sub-fórmula em $\Gamma \cup \Delta$, ou $s : (X \rightarrow B) \in \Delta$ ou $t : X \in \Delta$.

O nosso próximo passo, essencial para demonstrar a completude de $\mathcal{LP}$, consiste em provar a existência de uma extensão saturada, não derivável em $\mathcal{LPG}_0^-$, para cada sequência não derivável em $\mathcal{LPG}_0^-$.

[27] Alguns autores optam por incluir também, na definição de sequência saturada, as afirmações relativas aos restantes conectivos: Se $A \wedge B \in \Gamma$ então $A \in \Gamma$ e $B \in \Gamma$; se $A \wedge B \in \Delta$ então $A \in \Delta$ ou $B \in \Delta$; se $A \vee B \in \Gamma$ então $A \in \Gamma$ ou $B \in \Gamma$; se $A \vee B \in \Delta$ então $A \in \Delta$ e $B \in \Delta$; se $\neg A \in \Gamma$ então $A \in \Delta$ e se $\neg A \in \Delta$ então $A \in \Gamma$.

3.5.10 Lema (Lema da Saturação)

Consideremos uma sequência $\Gamma \Rightarrow \Delta$ tal que $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$. Então existe uma sequência saturada $\Gamma' \Rightarrow \Delta'$ tal que:

1. $\Gamma \subseteq \Gamma'$ e $\Delta \subseteq \Delta'$

2. $\Gamma' \Rightarrow \Delta'$ também não é derivável em $\mathcal{LPG}_0^-$, ou seja, $\mathcal{LPG}_0^- \nvdash \Gamma' \Rightarrow \Delta'$.

Demonstração

1. Comecemos por construir um Algoritmo de Saturação (AS) que nos permita obter a sequência saturada $\Gamma' \Rightarrow \Delta'$.

Seja $\Gamma \Rightarrow \Delta$ a sequência dada. Para cada fórmula, $S \in \Gamma \cup \Delta$, que ainda não tenha sido descarregada, tentemos levar a cabo um dos seguintes passos (1-7).

No momento inicial, 0, todas as fórmulas de $\Gamma \cup \Delta$ estão disponíveis.

Depois de efectuar o procedimento indicado num determinado passo, com respeito a uma fórmula S, essa fórmula é descarregada, ou seja, deixa de estar disponível para novos passos.

Se não for necessário levar a cabo nenhuma das instruções dos passos 1-7, então $\Gamma \Rightarrow \Delta$ verifica as condições de saturação e, nesse caso, terminamos o algoritmo com *sucesso*.

1 - Se $S = (A \rightarrow B) \in \Gamma$, então coloque A em Δ ou B em Γ.

2 - Se $S = (A \rightarrow B) \in \Delta$, então coloque A em Γ e B em Δ.

3 - Se $S = t : A \in \Gamma$, então coloque A em Γ.

4 - Se $S = !t : t : A \in \Delta$, então coloque $t : A$ em Δ.

5 - Se $S = (s + t) : A \in \Delta$, então coloque $s : A$ e $t : A$ em Δ.

6 - Se $S = st : B \in \Delta$, então, para cada X_i tal que $X_i \rightarrow B$ ocorra como sub-fórmula em $\Gamma \cup \Delta$, coloque $s : (X_i \rightarrow B)$ em Δ ou $t : X_i$ em Δ.

7 - Se $\Gamma \cap \Delta \neq \emptyset$ ou $\perp \in \Gamma$, então recue. Se recuar até ao primeiro passo, termine o algoritmo com falha. Sempre que recue a um dado passo, torne disponíveis todas as fórmulas que descarregou quando deixou esse passo pela última vez.

Enquanto percorremos o algoritmo, os conjuntos Γ e Δ vão sendo extendidos através da inclusão de novas fórmulas. Quando AS termina obtemos dois conjuntos Γ' e Δ', tais que $\Gamma' \supseteq \Gamma$ e $\Delta' \supseteq \Delta$.

- Verifiquemos que AS termina com sucesso.

AS <u>termina</u> pois tem um número finito de passos e, em cada passo em que não recue, descarrega ou uma fórmula de $\Gamma \Rightarrow \Delta$ ou uma fórmula do tipo $t : F$, onde t e F ocorrem ambos em $\Gamma \Rightarrow \Delta$. Havendo apenas um número finito destas fórmulas, o términus do algoritmo é garantido.

AS termina com <u>sucesso</u>, pois caso contrário AS terminaria no primeiro passo correspondente à sequência inicial $\Gamma \Rightarrow \Delta$, com todas as hipóteses esgotadas, incluindo a hipótese de recuar. Assim sendo, a sequência inicial $\Gamma \Rightarrow \Delta$ não satisfazia as hipóteses iniciais, verificando-se $\Gamma \cap \Delta \neq \emptyset$ ou $\perp \in \Gamma$.

- Provemos então que $\mathcal{LPG}_0^- \nvdash \Gamma' \Rightarrow \Delta'$. Suponhamos, com vista a um absurdo, que $\mathcal{LPG}_0^- \vdash \Gamma' \Rightarrow \Delta'$.

Notemos que após efectuado um determinado passo do algoritmo obtemos mais um nó da árvore. A cada nó corresponde uma sequência $\Pi \Rightarrow \Theta$, tal que, $\Gamma \subseteq \Pi \subseteq \Gamma'$ e $\Delta \subseteq \Theta \subseteq \Delta'$. A *árvore* descrita por este algoritmo, contém $\Gamma \Rightarrow \Delta$ na *raíz* e os axiomas de $\mathcal{LPG}_0$ nas *folhas*.

Por indução na *profundidade de um nó*, podemos provar que todas as sequências ($\Pi \Rightarrow \Theta$) da árvore são deriváveis em $\mathcal{LPG}_0^-$, incluindo a sequência de raíz $\Gamma \Rightarrow \Delta$, o que contradiz a hipótese $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$.

Os nós a que correspondem os passos de 1-5 e 7 são triviais. Em cada um dos passos 1-5 incluímos uma fórmula em Γ e/ou uma fórmula em Δ, não entrando em contradição com o que já existia na sequência anterior, e avançamos um passo. Logo, se a nova sequência é derivável também a anterior o era. No passo 7 recuamos um passo por forma a retirar da sequência as fórmulas que tenham dado origem a um dos casos $\Gamma \cap \Delta \neq \emptyset$ ou $\perp \in \Gamma$.

O passo crítico deste algoritmo é o passo 6 uma vez que a instrução correspondente poderia dar origem a uma das situações descritas no passo 7, obrigando à verificação posterior da não inclusão em Δ de fórmulas que já estavam em Γ. Verifiquemos, no entanto, que também nesse caso a sequência correspondente é derivável em $\mathcal{LPG}_0^-$.

Consideremos que chegámos a um nó que é relativo ao passo número 6. A este nó corresponde uma sequência do tipo $\Pi \Rightarrow \Theta \cup st : B$.

Suponhamos que $X_1, ...X_n$ são todas as fórmulas tal que $X_i \to B$, com $i = 1, ..., n$, é subfórmula de $\Gamma \cup \Delta$.

Neste caso, temos 2^n nós, descendentes deste, com sequências do tipo $\Pi \Rightarrow \Theta, st : B, Y_1^\sigma, Y_2^\sigma, ...Y_n^\sigma$ onde $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$ é um n-uplo de zeros e uns e

$$Y_i^\sigma = \begin{cases} s : (X_i \to B), \text{ se } \sigma_i = 0 \\ t : X_i, \text{ se } \sigma_i = 1 \end{cases}$$

(Temos tantos nós quantos os n-uplos de zeros e uns que é possível formar)

Por hipótese de indução, todos os descendentes são deriváveis em $\mathcal{LPG}_0^-$. Em particular, os 2^{n-1} pares de sequências do tipo $\Pi \Rightarrow \Theta', s : (X_1 \to B)$, sendo $\sigma_1 = 0$ e do tipo $\Pi \Rightarrow \Theta', t : X_1$, sendo $\sigma_1 = 1$.

Aplicando a regra ($\Rightarrow .$) a cada par de sequências deste tipo, obtemos $\Pi \Rightarrow \Theta', st : B$, ou seja, $\Pi \Rightarrow \Theta'$ (já que neste passo assumimos $st : B \in \Theta'$). Logo as 2^{n-1} sequências resultantes, do tipo $\Pi \Rightarrow \Theta'$, mais especificamente, $\Pi \Rightarrow \Theta, st : B, Y_2^\sigma, ...Y_n^\sigma$, são deriváveis em $\mathcal{LPG}_0^-$.

Repetindo este raciocínio $n-1$ vezes, terminamos com a derivabilidade da sequência $\Pi \Rightarrow \Theta, st : B$ em $\mathcal{LPG}_0^-$.

Deste modo, concluimos que todas as sequências da árvore são deriváveis em $\mathcal{LPG}_0^-$. Em particular, $\mathcal{LPG}_0^- \vdash \Gamma \Rightarrow \Delta$, contradizendo a hipótese do lema.

Logo, $\mathcal{LPG}_0^- \nvdash \Gamma' \Rightarrow \Delta'$. ■

3.5.11 Observação

Observemos que numa sequência saturada $\Gamma \Rightarrow \Delta$, não derivável em $\mathcal{LPG}_0^-$, Γ é fechado para as regras $\frac{t:A}{A}$ e $\frac{A \to B, A}{B}$.

3.5.12 Lema

Para cada sequência saturada $\Gamma \Rightarrow \Delta$, não derivável em $\mathcal{LPG}_0^-$, existe um conjunto de fórmulas de $\mathcal{LP}$, $\widetilde{\Gamma}$ (complemento de $\Gamma \Rightarrow \Delta$), tal que:

1. $\widetilde{\Gamma}$ é um conjunto decidível. Para cada termo t o conjunto $I(t) = \{X | t : X \in \widetilde{\Gamma}\}$ é finito e uma função que transforme um código de t[28] num código de $I(t)$[29] é computável.[30]

2. $\Delta \cap \widetilde{\Gamma} = \emptyset$. Se $F \in \Gamma$ então $F \in \widetilde{\Gamma}$.

3. Se $t : X \in \widetilde{\Gamma}$ então $X \in \widetilde{\Gamma}$.

4. Se $s : (X \to Y) \in \widetilde{\Gamma}$ e $t : X \in \widetilde{\Gamma}$ então $st : Y \in \widetilde{\Gamma}$.

5. Se $t : X \in \widetilde{\Gamma}$ então $!t : t : X \in \widetilde{\Gamma}$.

6. Se $t : X \in \widetilde{\Gamma}$ e s é um polinómio prova, então $(t+s) : X \in \widetilde{\Gamma}$ e $(s+t) : X \in \widetilde{\Gamma}$.

Demonstração

Vamos descrever um algoritmo COM que produz uma série de conjuntos finitos de fórmulas de $\mathcal{LP}$, Γ_0, Γ_1, ...

Seja $\Gamma_0 = \Gamma$.

Para cada número natural $i \geq 0$, COM define o conjunto Γ_{i+1} da seguinte forma:

Se $i = 3k$, para algum k natural, então:

$$\Gamma_{i+1} = \Gamma_i \bigcup_{s,t} \{s.t : Y | s : (X \to Y), t : X \in \Gamma_i\}$$

[28]Por exemplo o número de Gödel de t, $\#t$.

[29]Considerando todas as ordenações possíveis do conjunto finito de números de Gödel das fórmulas de $I(t)$, podemos determinar o número de Gödel de cada uma dessas sequências e tomar para código de $I(t)$, por exemplo, o menor desses números.

[30]Uma função $f : \mathbb{N} \to \mathbb{N}$ é computável sse existe um procedimento efectivo tal que, para cada $a \in \mathbb{N}$ produz $f(a)$.

Se $i = 3k + 1$, para algum k natural, então:

$$\Gamma_{i+1} = \Gamma_i \bigcup_t \{!t : t : X | t : X \in \Gamma_i\}$$

Se $i = 3k + 2$, para algum k natural, então:

$$\Gamma_{i+1} = \Gamma_i \bigcup_{s,t} \{(s+t) : X, (t+s) : X | t : X \in \Gamma_i, |s| < i^{31}\}$$

Tomemos $\widetilde{\Gamma} = \bigcup_i \Gamma_i$.

1. Pelo algoritmo, $\Gamma_i \subseteq \Gamma_{i+1}$, para cada número natural i. No passo $i \geq 0$, COM produz uma fórmula de Γ ou fórmulas do tipo $r : A$ com comprimento maior que $\frac{i}{3}$. Este facto garante a decidibilidade de $\widetilde{\Gamma}$ pois dada uma fórmula F, de comprimento n, no passo $i = 3n$ de COM: $F \in \Gamma_n$ sse $F \in \widetilde{\Gamma}$. Um argumento análogo estabelece a decidibilidade de $I(t)$. A partir do algoritmo ou procedimento efectivo associado à decidibilidade de $I(t)$ podemos construir a função computável de t para $I(t)$.

2. e 3. Provemos, por indução em i, as afirmações A,B e C que se seguem.

A. $\Gamma_i \cap \Delta = \emptyset$

B. Se $t : X \in \Gamma_i$ então $X \in \Gamma_i$

C. Se $X \to Y$, $X \in \Gamma_i$ então $Y \in \Gamma_i$

- Para $i = 0$, $\Gamma_0 = \Gamma$ e $\Gamma \Rightarrow \Delta$ é saturada, logo verifica-se A, B e C.
- Passo de indução

Por hipótese de indução, as afirmações A,B e C verificam-se para i.

A. Suponhamos que existe $F \in \Gamma_{i+1} \cap \Delta$. $F \notin \Gamma_i$ pois $\Gamma_i \cap \Delta = \emptyset$. Estudemos as três possibilidades.

Se $i = 3k$ então F é do tipo $st : Y$, com $s : (X \to Y), t : X \in \Gamma_i$, para algum X.

Da descrição de COM resulta que $X \to Y \in \Gamma = \Gamma_0$ (pois não foi introduzida em nenhum Γ_i posterior).

Como $st : Y \in \Delta$ (já que por hipótese $F \in \Gamma_{i+1} \cap \Delta$) e $X \to Y \in \Gamma$, então, pela saturação de $\Gamma \Rightarrow \Delta$, ou $s : (X \to Y) \in \Delta$ ou $t : X \in \Delta$. Logo, $s : (X \to Y) \in \Gamma_i \cap \Delta$ ou $t : X \in \Gamma_i \cap \Delta$. Em qualquer dos casos teríamos $\Gamma_i \cap \Delta \neq \emptyset$, contradizendo a hipótese de indução.

[31] Utilizamos $|s|$ para representar o comprimento do polinómio prova s, ou seja, o número de variáveis, constantes e símbolos funcionais existentes em s.

Se $i = 3k+1$ então F é do tipo $!t : t : X$ com $t : X \in \Gamma_i$. Pela propriedade (4) de saturação, $t : X \in \Delta$. Logo $t : X \in \Gamma_i \cap \Delta$ e novamente $\Gamma_i \cap \Delta \neq \emptyset$, contradizendo a hipótese de indução.

Se $i = 3k+2$ então F é do tipo $(t+s) : X$, com $t : X \in \Gamma_i$ ou $s : X \in \Gamma_i$. Como $(t+s) : X \in \Delta$, temos $t : X \in \Delta$ e $s : X \in \Delta$, pela propriedade (5) de saturação. Logo $t : X \in \Gamma_i \cap \Delta$ e $\Gamma_i \cap \Delta \neq \emptyset$.

Nos três casos a contradição deve-se à suposição de que existe $F \in \Gamma_{i+1} \cap \Delta$. Portanto, $\Gamma_{i+1} \cap \Delta = \emptyset$.

B. Suponhamos que $p : B \in \Gamma_{i+1}$ e $p : B \notin \Gamma_i$. (Se $p : B \in \Gamma_i$ é imediato, pois, por hipótese de indução, $B \in \Gamma_i$ e logo $B \in \Gamma_{i+1}$ já que $\Gamma_i \subseteq \Gamma_{i+1}$)

Estudemos novamente as três possibilidades.

Se $i = 3k$ então $p : B$ é do tipo $st : Y$, com $s : (X \rightarrow Y), t : X \in \Gamma_i$, para algum X. Por hipótese de indução, $X \rightarrow Y \in \Gamma_i$ e $X \in \Gamma_i$, logo $Y \in \Gamma_i$, por hipótese de indução (C). Como $\Gamma_i \subseteq \Gamma_{i+1}$, $Y \in \Gamma_{i+1}$, ou seja, $B \in \Gamma_{i+1}$.

Se $i = 3k+1$ então $p : B$ é do tipo $!t : (t : X)$ com $t : X \in \Gamma_i$. Logo $t : X \in \Gamma_{i+1}$, ou seja, $B \in \Gamma_{i+1}$.

Se $i = 3k+2$ então $p : B$ é do tipo $(t+s) : X$ com $|t+s| < i+1$, sendo que $t : X \in \Gamma_i$ e $|t| < i$ ou $s : X \in \Gamma_i$ e $|s| < i$.

Por hipótese de indução, $X \in \Gamma_i \subseteq \Gamma_{i+1}$, ou seja, $B \in \Gamma_{i+1}$.

C. Suponhamos que $X \rightarrow Y$, $X \in \Gamma_{i+1}$. Pela descrição de COM temos $X \rightarrow Y \in \Gamma$ (porque não foi incluída posteriormente, em nenhum dos três casos, num Γ_i). Pela saturação de $\Gamma \Rightarrow \Delta$, vem $Y \in \Gamma$ ou $X \in \Delta$. No primeiro caso é imediato que $Y \in \Gamma_{i+1}$. O segundo caso não pode ocorrer pois caso contrário $X \in \Gamma_i \cap \Delta$, contradizendo a hipótese de indução. Logo $Y \in \Gamma_{i+1}$.

Provadas as afirmações A,B e C,os pontos 2. e 3. do lema são imediatos:

2. $\Delta \cap \widetilde{\Gamma} = \emptyset$, pois $\Gamma_i \cap \Delta = \emptyset$ para todo o natural i. Se $F \in \Gamma = \Gamma_0$ então $F \in \widetilde{\Gamma}$.

3. Se $t : X \in \widetilde{\Gamma}$ então existe $i \geq 0$ tal que $t : X \in \Gamma_i$. Por **B**, $X \in \Gamma_i$ e logo $X \in \widetilde{\Gamma}$.

4. Suponhamos que $s : (X \rightarrow Y) \in \widetilde{\Gamma}$ e $t : X \in \widetilde{\Gamma}$. Seja j o menor número tal que $s : (X \rightarrow Y) \in \Gamma_j$ e k o menor número tal que $t : X \in \Gamma_k$. Seja m o maior entre eles. Então $s : (X \rightarrow Y), t : X \in \Gamma_m$. Tomemos um múltiplo de 3 maior ou igual a m, digamos i. Então $(s.t) : Y \in \Gamma_{i+1}$ e logo $(s.t) : Y \in \widetilde{\Gamma}$.

5. Consideremos $t : X \in \widetilde{\Gamma}$ e k o menor número tal que $t : X \in \Gamma_k$. Seja $i \geq k$ tal que $i = 3j + 1$, para algum j. Então $!t : t : X \in \Gamma_{i+1}$ e logo $!t : t : X \in \widetilde{\Gamma}$.

6. Finalmente suponhamos que $t : X \in \widetilde{\Gamma}$ e s é um polinómio prova. Seja k o menor número tal que $t : X \in \Gamma_k$. Tomemos $i \geq k$ tal que $i = 3j + 2$, para algum j. Então $(t + s) : X$ e $(s + t) : X$ pertencem a $\Gamma_{i+1} \subseteq \widetilde{\Gamma}$. ■

Demonstramos, em seguida, o teorema que nos vai permitir concluir a completude de $\mathcal{LP}$ com respeito à semântica proposta.

3.5.13 Teorema

As afirmações 1-5 são **equivalentes**.

1. $\mathcal{LPG}_0^- \vdash \Gamma \Rightarrow \Delta$.

2. $\mathcal{LPG}_0 \vdash \Gamma \Rightarrow \Delta$.

3. $\mathcal{LP}_0 \vdash \ \wedge \Gamma \rightarrow \vee \Delta$.

4. Para cada interpretação $*$, **AP** $\vdash$ $(\wedge \Gamma \rightarrow \vee \Delta)^*$.

5. Para cada interpretação $*$, a fórmula $(\wedge \Gamma \rightarrow \vee \Delta)^*$ é verdadeira.

Demonstração

O passo **1** implica **2** é trivial.

O passo **2** implica **3** foi demonstrado na proposição 1 desta secção.

O passo **3** implica **4** resulta da validade aritmética de $\mathcal{LP}_0$ (proposição 3.4.1).

O passo **4** implica **5** é trivial.

Resta demonstrar que **5** implica **1** para obter a equivalência entre todas as afirmações.

Por ser uma demonstração extensa, optamos por provar alguns resultados importantes inseridos em lemas e corolários.

Vamos utilizar um raciocínio por absurdo e provar que se não se verifica **1** então não se verifica **5**.

Suponhamos que $\mathcal{LPG}_0^- \nvdash \Gamma \Rightarrow \Delta$. Provemos que, nesse caso, existe uma interpretação $*$ para a qual $(\wedge \Gamma \rightarrow \vee \Delta)^*$ é falsa. O nosso objectivo passa, portanto, por construir uma tal interpretação.

Pelo Lema da Saturação, existe uma sequência saturada $\Gamma' \Rightarrow \Delta'$ tal que $\Gamma \subseteq \Gamma'$, $\Delta \subseteq \Delta'$ e $\mathcal{LPG}_0^- \nvdash \Gamma' \Rightarrow \Delta'$. Seja $\widetilde{\Gamma}'$ um complemento desta sequência, obtido através do algoritmo construído no Lema 3.5.12.

Vamos definir uma interpretação * das proposições S_i, das provas variáveis x_j e das provas constantes a_j, assumindo que a numeração de Gödel, da linguagem conjunta de $\mathcal{LP}$ e **AP**, é injectiva, ou seja, $\#E_1 = \#E_2 \leftrightarrow E_1 \equiv E_2$, para quaisquer expressões E_1 e E_2 (e que 0 não é número de Gödel de nenhuma expressão).

Tomemos então:

$$S^* = \#S = \begin{cases} \#S, \text{ se } S \in \widetilde{\Gamma}' \\ 0, \text{ se } S \notin \widetilde{\Gamma}' \end{cases}$$

$x^* = \#x$

$a^* = \#a$

Vejamos como extender esta interpretação às fórmulas.

Para cada fórmula aritmética do tipo $Prf(x, y)$ definimos uma translação $\dagger$ que transforma termos de $\mathcal{LP}$ em numerais e fórmulas de $\mathcal{LP}$ em fórmulas de **AP**, de tal forma que:

$S^\dagger = S^*$, para cada proposição S,

$t^\dagger = \#t$, para cada termo t,

$(t : F)^\dagger = Prf(t^\dagger, \#F^\dagger)$

e $\dagger$ comuta com os conectivos proposicionais.

Demonstra-se, por indução na construção da fórmula de $\mathcal{LP}$, que $\dagger$ é injectiva, isto é, se $F^\dagger = G^\dagger$ então $F \equiv G$. A demonstração baseia-se no seguinte facto: "Se $F^\dagger = G^\dagger$ então os principais conectivos de F e G coincidem" (**Artemov, 1998**).

Seja $(P, \otimes, \oplus, \Uparrow)$ o predicado prova aritmético natural. $P(x, y)$ significa que "x é o código de uma derivação da fórmula de código y" e $\otimes$, $\oplus$ e $\Uparrow$ as operações com provas (aplicação, escolha e verificador de provas, respectivamente) associadas a P. Em particular, para quaisquer fórmulas aritméticas φ, ψ e naturais k, n, as seguintes fórmulas são verdadeiras:

$$P\big(k, \#(\varphi \to \psi)\big) \wedge P(n, \#\varphi) \to P\big(k \otimes n, \#\psi\big)$$
$$P(k, \#\varphi) \to P\big(k \oplus n, \#\varphi\big)$$
$$P(n, \#\varphi) \to P\big(k \oplus n, \#\varphi\big)$$
$$P(k, \#\varphi) \to P\Big(\Uparrow k, \#\big(P(k, \#\varphi)\big)\Big)$$

Assumimos que $P(\#t, k)$ é falsa para qualquer termo t e $k \in \mathbb{N}$. (Podemos escolher um processo de godelização análogo ao descrito na nota de rodapé da página 43, por forma a que o código de um termo não seja código de uma derivação. No processo descrito, a decomposição em factores primos do número de Gödel de um termo tem o factor 2, enquanto que o código de uma derivação não, logo são diferentes.)

Vamos utilizar a notação $\mu x.\varphi(x,\vec{y})$ para a função que calcula x tal que $\varphi(x,\vec{y}) \wedge \forall z < x \neg\varphi(z,\vec{y})$, ou seja, $\mu x.\varphi(x,\vec{y})$ encontra o primeiro (menor) x que verifica $\varphi(x,\vec{y})$.

Ora, $\mu x.\varphi(x,\vec{y})$ é computável se a condição $\varphi(x,\vec{y}) \wedge \forall z < x \neg\varphi(z,\vec{y})$ é Σ_1-demonstrável. Portanto, cada uma das condições seguintes é suficiente para a função ser computável:

(C1) $\varphi(x,\vec{y})$ é Δ_1-demonstrável[32]

(C2) $\varphi(x,\vec{y})$ é Σ_1-demonstrável e é função injectiva com respeito à variável x, ou seja, $\varphi(k_1,\vec{n}) \wedge \varphi(k_2,\vec{n}) \rightarrow k_1 = k_2$ é verdadeira, quaisquer que sejam k_1, k_2 e $\vec{n}$.

Vamos definir a fórmula $Prf(x,y)$ de tal forma que **AP** prova a seguinte **equação de ponto fixo (EPF)**:

$$Prf(x,y) \leftrightarrow P(x,y) \vee (\text{“}x = \#t, \text{ para algum termo } t \text{ de } \mathcal{LP} \text{ e } y = \#B^{\dagger}, \text{ para alguma fórmula } B \text{ de } \mathcal{LP}, \text{ tal que } B \in I(t)\text{”})$$

Observação: Segundo esta definição, $Prf(x,y)$ é verdadeira se ocorrer uma de duas situações: ou x é o código de uma prova da fórmula de código y ou x é o número de Gödel de um termo t de $\mathcal{LP}$ e y o número de Gödel da fórmula aritmética que é interpretação de uma fórmula B de $\mathcal{LP}$ tal que $t : B \in \widetilde{\Gamma}'$.[33]

Notemos que a fórmula aritmética dentro de aspas descreve um procedimento primitivo recursivo que, dados x e y encontra t e B tais que $x = \#t$ e $y = \#B^{\dagger}$, verificando depois se $B \in I(t)$.

De (EPF) concluímos que Prf é Δ_1-demonstrável (já verificámos que $P(x,y)$ o é) e que, se **AP** $\vdash \varphi$ então $Prf(k,\#\varphi)$, para algum k (pois sendo k o número de Gödel da derivação, temos $P(k,\#\varphi)$ e logo, por EPF, $Prf(k,\#\varphi)$.

Para definir as funções computáveis m, a e c, associadas a Prf, vamos definir fórmulas $M(x,y,z)$, $A(x,y,z)$ e $C(x,z)$, da seguinte forma:

[32] Nesse caso φ e $\neg\varphi$ são Σ_1-demonstráveis. Logo, segundo a definição de G. Boolos (ver nota de rodapé n.º18), $\forall z < x \neg\varphi$ é Σ_1-demonstrável e $\varphi \wedge \forall z < x \neg\varphi$ é Σ_1-demonstrável.

[33] Recordemos que nem todos os número naturais são números de Gödel de fórmulas. Por exemplo, segundo o processo de Gödelização descrito na nota de rodapé n.º13, 5 e 9 não são números de Gödel de nenhuma fórmula ou dedução. Como 0 não é número de Gödel, 5 é apenas o código do símbolo “(” e 9 o código do símbolo $=$ ou o código da sequência $\neg$ (que não é uma fórmula nem dedução).

$$M(x,y,z) \leftrightarrow \Big(\text{“existem termos } s \text{ e } t\text{, de } \mathcal{LP}\text{, tais que } x = \#s \text{ e } y = \#t\text{”} \wedge z = \#st\Big) \bigvee$$

$$\Big(\text{“}x = \#s\text{, para algum termo } s \text{ de } \mathcal{LP}\text{, e } y \neq \#t\text{, qualquer que seja o termo } t \text{ de } \mathcal{LP}\text{”}$$

$$\bigwedge \exists v[\text{“}v = \mu\omega.\Big(\wedge \{P(\omega, \#B^{\dagger}) | B \in I(s)\}\Big)\text{”} \bigwedge z = v \otimes y]\Big) \bigvee$$

$$\Big(\text{“}x \neq \#s\text{, qualquer que seja o termo } s \text{ de } \mathcal{LP}\text{, e } y = \#t\text{, para algum termo } t \text{ de } \mathcal{LP}\text{”}$$

$$\bigwedge \exists v[\text{“}v = \mu\omega.\Big(\wedge \{P(\omega, \#B^{\dagger}) | B \in I(t)\}\Big)\text{”} \bigwedge z = x \otimes v]\Big) \bigvee$$

$$\Big(\text{“}x \neq \#s \text{ e } y \neq \#t\text{, quaisquer que sejam os termos } s \text{ e } t \text{ de } \mathcal{LP}\text{”} \bigwedge z = x \otimes y\Big)$$

$$A(x,y,z) \leftrightarrow \Big(\text{“existem termos } s \text{ e } t\text{, de } \mathcal{LP}\text{, tais que } x = \#s \text{ e } y = \#t\text{”} \wedge z = \#(s+t)\Big) \bigvee$$

$$\Big(\text{“}x = \#s\text{, para algum termo } s \text{ de } \mathcal{LP}\text{, e } y \neq \#t\text{, qualquer que seja o termo } t \text{ de } \mathcal{LP}\text{”}$$

$$\bigwedge \exists v[\text{“}v = \mu\omega.\Big(\wedge \{P(\omega, \#B^{\dagger}) | B \in I(s)\}\Big)\text{”} \bigwedge z = v \oplus y]\Big) \bigvee$$

$$\Big(\text{“}x \neq \#s\text{, qualquer que seja o termo } s \text{ de } \mathcal{LP}\text{, e } y = \#t\text{, para algum termo } t \text{ de } \mathcal{LP}\text{”}$$

$$\bigwedge \exists v[\text{“}v = \mu\omega.\Big(\wedge \{P(\omega, \#B^{\dagger}) | B \in I(t)\}\Big)\text{”} \bigwedge z = x \oplus v]\Big) \bigvee$$

$$\Big(\text{“}x \neq \#s \text{ e } y \neq \#t\text{, quaisquer que sejam os termos } s \text{ e } t \text{ de } \mathcal{LP}\text{”} \bigwedge z = x \oplus y\Big)$$

$$C(x,z) \leftrightarrow \Big(\text{“}x = \#t\text{, para algum termo } t \text{ de } \mathcal{LP}\text{”} \bigwedge z = \#!t\Big) \bigvee$$

$$\Big(\text{“}x \neq \#t\text{, qualquer que seja o termo } t \text{ de } \mathcal{LP}\text{”} \bigwedge$$

$$\exists v[\text{“}v = \mu\omega.\Big(\wedge \{P\big(\omega, \#P(t,\#\varphi) \rightarrow Prf(t,\#\varphi)\big) | \varphi \in T(t)\}\Big)\text{”} \bigwedge z = v \Uparrow x]\Big)$$

Comentários:

Para definir cada uma das fórmulas anteriores, tivemos que estudar separadamente os casos possíveis que poderiam ocorrer com os números x e y. Tomemos, a título de exemplo, a definição da primeira fórmula (M) cujo objectivo é obter um número z que poderá ser o resultado de m aplicado a x e y. Suponhamos então que x e y são códigos de provas de $F \rightarrow G$ e F, respectivamente, e queremos obter o código de G.

1. Se x e y são números de Gödel de termos de $\mathcal{LP}$, temos o problema resolvido devido à operação aplicação do sistema $\mathcal{LP}$ (def. 3.2.6, A_2).

2. Se x e y são códigos de provas de fórmulas aritméticas, também temos o problema resolvido devido à operação $\otimes$ associada ao predicado aritmético P.

3. Se um dos números é o código de uma prova de uma fórmula aritmética e o outro é o código de um termo t de $\mathcal{LP}$, então através da translação † , transformamos todas as fórmulas B de $\mathcal{LP}$, que verificam $t : B$, em fórmulas aritméticas e determinamos um código de uma prova dessa fórmula aritmética, reduzindo então este caso ao caso anterior.

A explicação do raciocínio utilizado em cada uma das restantes definições é análoga.

Tomando o predicado prova normal, *Prf*, definido através de EPF, resta definir as funções m, a e c, para concluir a definição da interpretação aritmética *. Sejam:

$$m(x,y) := \mu z.M(x,y,z)$$
$$a(x,y) := \mu z.A(x,y,z)$$
$$c(x) := \mu z.C(x,z)$$

As funções m, a e c são computáveis, pelas propriedades de M, A e C.[34]

Nos seguintes passos da demonstração provamos resultados importantes que enunciamos através de alguns lemas e respectivos corolários.

3.5.14 Lema

a) $t^* = t^\dagger$, qualquer que seja o termo t de $\mathcal{LP}$.

b) $B^* \equiv B^\dagger$, qualquer que seja a fórmula B de $\mathcal{LP}$.

Demonstração

a) Demonstremos esta igualdade por indução na construção do termo t.

No caso de t ser uma variável ou uma constante, a igualdade resulta da definição de *. Consideremos os casos em que t é $r.s$, $r + s$ ou $!s$. Por hipótese de indução, $r^* = r^\dagger$ e $s^* = s^\dagger$. Logo, pelas definições anteriores:

$$(r.s)^* = m(r^*, s^*) = m(\#r, \#s) = \#(r.s) = (r.s)^\dagger$$
$$(r+s)^* = a(r^*, s^*) = a(\#r, \#s) = \#(r+s) = (r+s)^\dagger$$
$$(!s)^* = c(s^*) = c(\#s) = \#(!s) = (!s)^\dagger$$

b) Demonstremos a identidade por indução na construção da fórmula B.

O caso em que B é uma fórmula atómica, ou seja, uma proposição, resulta da definição de *. Consideremos em seguida os restantes casos.

Suponhamos que B é do tipo $t : F$. Por hipótese de indução, $F^* = F^\dagger$. Logo, aplicando a):

$(t : F)^* = Prf(t^*, \#F^*) = Prf(t^\dagger, \#F^\dagger) = (t : F)^\dagger$

Suponhamos agora que B é de um dos tipos $F \to G$, $F \wedge G$, $F \vee G$ ou $\neg F$. Por hipótese de indução, $F^* = F^\dagger$ e $G^* = G^\dagger$. Como * e † comutam com os conectivos proposicionais, temos, para cada caso:

[34]Pelas definições de M, A e C, nas funções $\mu w.\psi$, ψ é sempre Δ_1-demonstrável. Logo, por (C1) são funções computáveis. Sendo M, A e C fórmulas Σ_1-demonstráveis e injectivas (com respeito a z) então m, a e c são computáveis por (C2).

$$(F \to G)^* = (F^* \to G^*) = (F^\dagger \to G^\dagger) = (F \to G)^\dagger$$
$$(F \wedge G)^* = (F^* \wedge G^*) = (F^\dagger \wedge G^\dagger) = (F \wedge G)^\dagger$$
$$(F \vee G)^* = (F^* \vee G^*) = (F^\dagger \vee G^\dagger) = (F \vee G)^\dagger$$
$$(\neg F)^* = \neg F^* = \neg F^\dagger = (\neg F)^\dagger$$

3.5.15 Corolário

A aplicação * é injectiva com respeito aos termos e fórmulas de $\mathcal{LP}$. Em particular, para quaisquer expressões E_1 e E_2, temos $E_1{}^* = E_2{}^* \Rightarrow E_1 \equiv E_2$.

Demonstração

A injectividade de * resulta da, já referida, injectividade de †, já que:

$E_1{}^* = E_2{}^* \Rightarrow E_1^\dagger = E_2^\dagger \Rightarrow E_1 \equiv E_2$

3.5.16 Corolário

Para toda a fórmula X de $\mathcal{LP}$, X^* é Δ_1-demonstrável.

Demonstração

Se X é atómica, $X = S$. Por definição, S^* é zero ou um número de Gödel, logo equivalente a $\exists x(x = 0)$ ou a $\exists x(x = \#S)$. Sendo $x = 0$ e $x = \#S$ Σ_1-demonstráveis, X^* é Δ_1-demonstrável.

Se X é do tipo $t : F$, então $(t : F)^* = \mathit{Prf}(t^*, \#F^*)$. Pelo lema 3.5.14, $\mathbf{AP} \vdash \mathit{Prf}(t^*, \#F^*) \leftrightarrow \mathit{Prf}(\#t, \#F^*)$. Ora, como $\mathit{Prf}(\#t, \#F^*)$ é Δ_1-demonstrável, também $(t : F)$ o é.

Se X é de um dos tipos $F \to G$, $F \wedge G$, $F \vee G$ ou $\neg F$, o resultado segue do facto da classe de fórmulas Δ_1-demonstráveis ser fechada para os conectivos booleanos.

3.5.17 Lema

Seja X uma fórmula de $\mathcal{LP}$.

a) Se $X \in \widetilde{\Gamma}'$, então $\mathbf{AP} \vdash X^*$.

b) Se $X \in \Delta'$, então $\mathbf{AP} \vdash \neg X^*$.

Demonstração

Mais uma vez demonstremos o lema por indução na complexidade de X.

- Provemos primeiro os casos básicos.

Se X é atómica, ou seja, $X = S$ para alguma proposição S, então X^* é verdadeira[35] se e só se $\#X \neq 0$, ou seja, se e só se $X \in \widetilde{\Gamma}'$.

Se $X \in \Delta'$, $\#X = 0$, então X não é fórmula derivável em **AP** (é falsa).

Suponhamos agora que $X = t : F$ para algum termo t e fórmula F.

[35]Pelo corolário anterior, se X^* é verdadeira então é derivável.

a) Se $t : F \in \widetilde{\Gamma}'$, então **AP** $\vdash$ "$F \in I(t)$", por definição de complemento de $\widetilde{\Gamma}$. Por EPF, **AP** $\vdash$ $Prf(\#t,\#F^{\dagger})$ e pelo lema 3.5.14 **AP** $\vdash$ $Prf(\#t,\#F^*)$, ou seja, **AP** $\vdash$ $(t : F)^*$.

b) Se $t : F \in \Delta'$, então $t : F \notin \widetilde{\Gamma}'$ e "$F \in I(t)$" é falsa. A fórmula $P(t^*,\#F^*)$ também é falsa pois assumimos $P(\#t, k)$ falsa para todo o k.

Então, por EPF, $(t : F)^*$ é falsa (pois não temos $P(t^*,\#F^*)$ nem $F \in I(t)$). Como $(t : F)^*$ é Δ_1-demonstrável (pelo corolário 3.5.16), então **AP** $\vdash \neg(t : F)^*$.

- Passo de indução

Suponhamos que X é de um dos tipos $F \to G$, $F \wedge G$, $F \vee G$ ou $\neg F$. Por hipótese de indução, F e G verificam a) e b). O resultado segue das propriedades de saturação de $\Gamma' \Rightarrow \Delta'$

Caso $X = F \to G$.

a) Se $F \to G \in \widetilde{\Gamma}'$, então $F \to G \in \Gamma'$. Pela propriedade de saturação de Γ', $G \in \Gamma'$ ou $F \in \Delta'$. Por hipótese de indução, **AP** $\vdash G^*$ ou **AP** $\vdash \neg F^*$, ou seja, G^* é verdadeira ou F^* é falsa. Em qualquer dos casos, $(F \to G)^*$ é verdadeira, pelo que, **AP** $\vdash (F \to G)^*$.

b) Se $X = F \to G \in \Delta'$, então $F \in \Gamma'$ e $G \in \Delta'$. Por hipótese de indução, **AP** $\vdash F^*$ e **AP** $\vdash \neg G^*$, ou seja, F^* é verdadeira e G^* é falsa. Logo, $(F \to G)^*$ é falsa e **AP** $\vdash \neg(F \to G)^*$.

Caso $X = F \wedge G$.

a) Se $F \wedge G \in \widetilde{\Gamma}'$, então $F \in \widetilde{\Gamma}'$ e $G \in \widetilde{\Gamma}'$. Por hipótese de indução, **AP** $\vdash F^*$ e **AP** $\vdash G^*$, logo **AP** $\vdash F^* \wedge G^*$, ou seja, **AP** $\vdash (F \wedge G)^*$.

b) Se $F \wedge G \in \Delta'$, então $F \in \Delta'$ ou $G \in \Delta'$. Por hipótese de indução, **AP** $\vdash \neg F^*$ ou **AP** $\vdash \neg G^*$, logo F^* ou G^* são falsas, pelo que $F^* \wedge G^*$ é falsa. Portanto, **AP** $\vdash \neg(F \wedge G)^*$.

Caso $X = F \vee G$.

a) Se $F \vee G \in \widetilde{\Gamma}'$, então $F \in \widetilde{\Gamma}'$ ou $G \in \widetilde{\Gamma}'$. Por hipótese de indução, **AP** $\vdash F^*$ ou **AP** $\vdash G^*$, logo **AP** $\vdash F^* \vee G^*$, ou seja, **AP** $\vdash (F \vee G)^*$.

b) Se $F \vee G \in \Delta'$, então $F \in \Delta'$ e $G \in \Delta'$. Por hipótese de indução, **AP** $\vdash \neg F^*$ e **AP** $\vdash \neg G^*$, pelo que F^* é falsa e G^* é falsa. Logo, $F^* \vee G^*$ é falsa e portanto **AP** $\vdash \neg(F \vee G)^*$.

Caso $X = \neg F$.

a) Se $\neg F \in \widetilde{\Gamma}'$, então $F \notin \widetilde{\Gamma}'$ e $F \in \Delta'$. Por h.i., **AP** $\vdash \neg F^*$, ou seja, **AP** $\vdash (\neg F)^*$.

b) Se $\neg F \in \Delta'$, então $F \notin \Delta'$ e $F \in \widetilde{\Gamma}'$. Por h.i., **AP** $\vdash F^*$, logo F^* é verdadeira. Então $\neg F^*$ é falsa, ou seja, **AP** $\vdash \neg(\neg F)^*$.

3.5.18 Lema

AP $\vdash \varphi \Leftrightarrow Prf(n, \#\varphi)$ para algum $n \in \mathbb{N}$.

Demonstração

$\Rightarrow$ imediato

$\Leftarrow$ Seja n tal que $Prf(n,\#\varphi)$. Por EPF, temos $P(n,\#\varphi)$ ou $\#\varphi = \#B^{\dagger}$ para algum B tal que $t : B \in \widetilde{\Gamma}'$ (t é tal que $n = \#t$)

No primeiro caso, n é o código de uma derivação da fórmula φ de código $\#\varphi$, logo **AP** $\vdash \varphi$.

No segundo caso, pela saturação de $\widetilde{\Gamma}'$, $B \in \widetilde{\Gamma}'$. Pelo lema anterior (3.5.17) **AP** $\vdash B^*$.

Ora, pela injectividade da numeração de Gödel, $\varphi \equiv B^{\dagger}$, ou seja, $\varphi \equiv B^*$.

Portanto, **AP** $\vdash \varphi$.

3.5.19 Lema

Para quaisquer fórmulas aritméticas φ, ψ e números naturais k e n:

a) $Prf(k,\#(\varphi\rightarrow\psi)) \wedge Prf(n,\#\varphi) \rightarrow Prf(m(k,n),\#\psi)$

b) $Prf(k,\#\varphi) \rightarrow Prf(a(k,n),\#\varphi)$ e $Prf(n,\#\varphi) \rightarrow Prf(a(k,n),\#\varphi)$

c) $Prf(k,\#\varphi) \rightarrow Prf\Big(c(k),\#\big(Prf(k,\#\varphi)\big)\Big)$

Demonstração

a) Suponhamos que $Prf(k,\#(\varphi\rightarrow\psi))$ e $Prf(n,\#\varphi)$.

Estudemos as quatro possibilidades:

i) Nenhum dos números k e n é número de Gödel de um termo de $\mathcal{LP}$.

Por EPF, vem $P(k,\#(\varphi\rightarrow\psi))$ e $P(n,\#\varphi)$. Logo, $P(k\otimes n,\#\psi)$, ou seja, $Prf(m(k,n),\#\psi)$, novamente por EPF e por definição de m.

ii) k e n são números de Gödel dos termos s e t, respectivamente.

Por EPF, existem fórmulas F e G de $\mathcal{LP}$ tal que:

$\varphi = F^*$, $\psi = G^*$, $F \in I(t)$ e $F\rightarrow G \in I(s)$

Pela propriedade 4 de saturação de $\widetilde{\Gamma}'$(lema 3.5.12), $(s.t) : G \in \widetilde{\Gamma}'$, ou seja, $G \in I(s.t)$.

Por EPF, $Prf(\#(st),\#G^*)$. No entanto, pelo lema 3.5.17 e pelas definições:

$\#(st) = (st)^* = m(s^*,t^*) = m(\#s,\#t) = m(k,n)$

Logo $Prf(m(k,n),\#\psi)$ é verdadeira.

iii) k não é número de Gödel de nenhum termo de $\mathcal{LP}$, mas n é o número de Gödel de um termo t, ou seja, $n = \#t$.

Por EPF, $P(k,\#(\varphi\rightarrow\psi))$ e existe uma fórmula F de $\mathcal{LP}$ tal que $\varphi \equiv F^{\dagger}$ com $F \in I(t)$.

Determinemos o número $j = \mu\omega.\Big(\wedge \{P(\omega,\#B^{\dagger}) | B \in I(t)\}\Big)$ através do seguinte método.

Seja $I(t) = \{B_1,...,B_l\}$. Por definição, $B_i \in \widetilde{\Gamma}'$, para todo $i = 1,...,l$. Sendo assim, **AP** $\vdash B_i^{\dagger}$, para todo $i = 1,...l$, por aplicação dos lemas 3.5.17 e 3.5.14.

Pela propriedade 2 do predicado prova P existe ω tal que $P(\omega, \#B_i^\dagger)$, para todo $i = 1, ..., l$. Seja j o menor dos ω nestas condições.

Em particular, temos $P(j, \#F^\dagger)$.

De $P(k, \#(\varphi \to \psi))$ e $P(j, \#\varphi)$ concluímos $P(k \otimes j, \#\psi)$, por definição de $\otimes$.

Por definição de m e M, temos $m(k, n) = k \otimes j$.

Portanto, $Prf(m(k, n), \#\psi)$ é verdadeira.

iv) n não é número de Gödel de nenhum termo de $\mathcal{LP}$, mas k é o número de Gödel de um termo s, ou seja, $k = \#s$.

A demonstração deste caso é perfeitamente análoga à do caso anterior (iii).

b) Suponhamos que $Prf(k, \#\varphi)$ ou $Prf(n, \#\varphi)$.

Estudemos novamente as quatro possibilidades:

i) Nenhum dos números k e n é número de Gödel de um termo de $\mathcal{LP}$.

Por EPF, vem $P(k, \#\varphi)$ ou $P(n, \#\varphi)$. Em ambos os casos, $P(k \oplus n, \#\varphi)$, ou seja, $Prf(a(k, n), \#\varphi)$, novamente por EPF e por definição de a.

ii) k e n são números de Gödel dos termos s e t, respectivamente.

Por EPF, existe uma fórmula F de $\mathcal{LP}$, tal que $\varphi \equiv F^*$ e $F \in I(s) \cap I(t)$.

Pela propriedade 6, de saturação de $\widetilde{\Gamma}'$ (lema 3.5.12), $(t + s) : F \in \widetilde{\Gamma}'$ e $(s + t) : F \in \widetilde{\Gamma}'$, ou seja, $F \in I(s + t) \cap I(t + s)$.

Novamente por EPF, $Prf(\#(s + t), \#F^*)$.

Como $\#(s + t) = (s + t)^* = a(s^*, t^*) = a(\#s, \#t) = a(k, n)$, então $Prf(a(k, n), \#\varphi)$.

iii) k não é número de Gödel de nenhum termo de $\mathcal{LP}$, mas n é o número de Gödel de um termo t, ou seja, $n = \#t$.

Por EPF, $P(k, \#\varphi)$ ou existe uma fórmula F de $\mathcal{LP}$, tal que $\varphi \equiv F^\dagger$ com $F \in I(t)$.

No primeiro caso é imediato que $P(k \oplus n, \#\varphi)$, logo $Prf(a(k, n), \#\varphi)$.

No segundo caso, determinemos o número $j = \mu\omega.\left(\wedge \{P(\omega, \#B^\dagger) | B \in I(t)\}\right)$ como em a.iii), de tal modo que $P(j, \#F^\dagger)$, ou seja, $P(j, \#\varphi)$. Logo, pelas propriedades de $\oplus$, $P(k \oplus j, \#\varphi)$.

Por definição de a e A, temos $k \oplus j = a(k, n)$.

Portanto, $Prf(a(k, n), \#\varphi)$ é verdadeira.

iv) n não é número de Gödel de nenhum termo de $\mathcal{LP}$, mas k é o número de Gödel de um termo s, ou seja, $k = \#s$.

A demonstração deste caso é perfeitamente análoga à do caso anterior (iii).

c) Suponhamos agora que $Prf(k, \#\varphi)$.

Neste caso há apenas duas possibilidades a estudar.

i) k é o número de Gödel de um termo de $\mathcal{LP}$, digamos t.

Por EPF, existe F, tal que $\varphi \equiv F^{\dagger}$ e $F \in I(t)$. Mas, pela propriedade 5, de saturação de $\widetilde{\Gamma}'$ (lema 3.5.12), $!t : t : F \in \widetilde{\Gamma}'$ e logo, pelo lema 3.5.17, **AP** $\vdash (!t : t : F)^*$.

Como $(!t : t : F)^* \equiv Prf(c(t^*), \#Prf(t^*, \#F^*))$ e $t^* = \#t = k$, $F^* \equiv F^{\dagger} \equiv \varphi$, então $Prf(c(k), \#Prf(k, \#\varphi))$ é verdadeira.

ii) k não é número de Gödel de nenhum termo de $\mathcal{LP}$.

Por EPF, $P(k, \#\varphi)$ e logo, por definição de $\Uparrow$, vem $P(\Uparrow k, \#P(k, \#\varphi))$ (1).

Tomemos $j = \mu\omega.\left(\bigwedge\{P(\omega, \#[P(k, \#\psi) \rightarrow Prf(k, \#\psi)]) \mid \psi \in T(k)\}\right)$.

Em particular, $P(j, \#[P(k, \#\varphi) \rightarrow Prf(k, \#\varphi)])$ (2).

De (1) e (2) concluímos $P(j \otimes \Uparrow k, \#Prf(k, \#\varphi))$.

Logo, por EPF, $Prf(j \otimes \Uparrow k, \#Prf(k, \#\varphi))$ e como $c(k) = j \otimes \Uparrow k$, pela definição de c, $Prf(c(k), \#Prf(k, \#\varphi))$.

3.5.20 Lema

Prf é um predicado normal.

Demonstração

Por EPF, *Prf* é Δ_1-demonstrável.

No lema 3.5.18, verificámos que, para qualquer sentença aritmética φ:

AP $\vdash \varphi$ sse $Prf(n, \#\varphi)$ é verdadeira para algum natural n.

1. Provemos que *Prf* verifica a primeira condição de normalidade.

Consideremos um número natural qualquer.

Se k é o número de Gödel de um termo t de $\mathcal{LP}$, então $T(k)$ é finito pois $I(k)$ é finito.

Se k não é número de Gödel de nenhum termo de $\mathcal{LP}$, então o resultado segue da normalidade do predicado aritmético P, já que podemos encontrar um algoritmo para P e construir um algoritmo para a função que transforma k no conjunto de numerais $T(k)$, através do algoritmo de decisão COM do lema 3.5.13.

Então, para cada prova k, o conjunto $T(k) = \{l : Prf(k, l)\}$ é finito.

2. Provemos que *Prf* verifica a segunda condição de normalidade.

Consideremos dois números naturais k e n quaisquer.

Pelo lema 3.5.19, $T(k) \cup T(n) \subseteq T(a(k, n))$. Tomemos $m = a(k, n)$. Então, existe um número natural m tal que $T(k) \cup T(n) \subseteq T(m)$.

Finalmente, terminemos a demonstração do Teorema.

Construímos uma interpretação * tal que, pelo lema 3.5.17, Γ^* é verdadeiro e Δ^* é falso (no modelo aritmético standard). Logo $(\bigwedge\Gamma \rightarrow \bigvee\Delta)^*$ é falsa.

Sendo assim **5** $\Rightarrow$ **1** e as cinco afirmações são equivalentes. ∎

3.5.21 Corolário

$\mathcal{LP}_0$ é decidível.

Demonstração

Dada uma fórmula F de $\mathcal{LP}_0$, apliquemos o algoritmo de saturação AS (lema 3.5.10), à sequência $\Rightarrow F$, de antecedente vazio.

Se AS falha então $\mathcal{LP}_0 \vdash F$. Se AS termina com sucesso então $\mathcal{LP}_0 \nvdash F$. ■

O nosso objectivo inicial é agora facilmente atingido, pois é uma consequência quase imediata do Teorema 3.5.13.

3.5.22 Corolário (Completude de $\mathcal{LP}$)

$\mathcal{LP}$(EC) $\vdash F$ sse **AP** $\vdash F^*$, qualquer que seja a interpretação-EC.

Demonstração

Notemos que a implicação $\Rightarrow$ já foi demonstrada no Corolário 3.4.1 (validade aritmética de $\mathcal{LP}$).

No entanto, pelo Teorema anterior, utilizando novamente a sequência $\Rightarrow F$, obtemos:

$\mathcal{LPG}_0^- \vdash \Rightarrow F \Leftrightarrow$

$\mathcal{LP}_0 \vdash F \Leftrightarrow$

$\mathcal{LP}$(EC) $\vdash F \Leftrightarrow$

Para cada interpretação $*$, **AP** $\vdash F^* \Leftrightarrow$

Para cada interpretação $*$, a fórmula F^* é verdadeira.

O resultado segue da terceira equivalência. ■

3.6 Formalização de BHK

Segundo Artemov (1999), *a priori*, os requisitos mínimos de uma semântica para a lógica intuicionista que formalize BHK são:

1. Basear-se em provas reais numa determinada teoria formal T (ou classe de teorias) que lhe sirva de fundamento. Em particular:

a) O predicado $P(p, F)$ deve ser decidível

b) As provas de BHK devem enumerar os teoremas da teoria, ou seja, $T \vdash F$ sse $P(p, F)$ para algum p.

2. Ser "não circular", ou seja, as provas de BHK não devem ser derivações num sistema formal baseado na própria lógica intuicionista.

Em 1986, Dirk van Dalen escreve:

"The intended interpretation of intuitionistic logic as presented by Heyting, Kreisel and others [BHK] so far has proved to be rather elusive..." **(Artemov, 1999)**

Nesta secção faremos um esboço da demonstração de que a semântica proposicional BHK (clássica[36]) admite uma formalização matemática exacta que especifica *Int*[37] baseada na noção clássica usual de prova, independentemente de quaisquer pressupostos intuicionistas.

Gödel em 1933 introduziu o chamado *cálculo modal de demonstrabilidade* e definiu *Int* nessa lógica. Gödel utiliza como operador modal de demonstrabilidade, o conectivo unário $\Box$ com a interpretação informal "F é demonstrável". A lógica de demonstrabilidade de Gödel inclui todos os axiomas e regras da lógica clássica e os seguintes axiomas e regra modais:

$\Box F \to F$

$\Box(F \to G) \to (\Box F \to \Box G)$

$\Box F \to \Box\Box F$

$F \vdash \Box F$ (regra de necessidade)

Os axiomas e a regra modal apresentada coincidem com os da lógica modal S4 **(Artemov, 2004)**.

3.6.1 Definição

Denotemos por $gk(F)$ a **translação** de Gödel-kolmogorov de uma fórmula de *Int* na lógica modal, que coloca o prefixo $\Box$ antes de todas as sub-fórmulas que ocorrem em F.

Gödel estabeleceu que $Int \vdash F \Rightarrow \text{S4} \vdash gk(F)$, encontrando uma leitura das fórmulas de *Int* em termos de afirmações sobre demonstrabilidade na matemática clássica. A equivalência seria estabelecida posteriormente, em 1948, por McKinsey e Tarski. Em 1938, Gödel lançou o problema de encontrar uma semântica de demonstrabilidade para S4[38], por forma a completar o esquema iniciado:

$$Int \hookrightarrow \text{S4} \hookrightarrow \ldots ? \ldots \hookrightarrow \text{“Provas reais”}$$

Entenda-se por "provas reais" um sistema formal e completo de provas.

Desde então, foram muitos os matemáticos que se dedicaram à procura da semântica desejada para S4. No entanto, a questão do cumprimento dos requisitos mínimos atrás

[36]Recordar as diferentes abordagens referidas na secção 1.2.

[37]Utilizamos *Int* como abreviatura para a lógica proposicional intuicionista. Uma outra abreviautra utilizada por diversos autores é IPC.

[38]A leitura imediata de $\Box$ como "F é demonstrável" contradizia o teorema de incompletude de Gödel.

referidos, levou a que o problema de encontrar a semântica adequada permanecesse em aberto.

A principal dificuldade tinha origem na aplicação do quantificador existencial a provas na fórmula $\exists x P(x,F)$[39]. O processo utilizado para contornar essa dificuldade consistia em trocar as fórmulas do tipo $\exists x P(x,F)$ por fórmulas $P(t,F)$, substituindo o quantificador existencial por operações sobre provas. No entanto, a ideia apenas se tornou exequível com Artemov, por ser suportada por uma linguagem apropriada (que inclui a operação !) e um sistema completo de axiomas, como revelou ser $\mathcal{LP}$.

O Teorema de completude demonstrado na secção 3.5 traduz o facto de $\mathcal{LP}$ conter todas as verdades lógicas, o que, na notação utilizada anteriormente, significa que foi encontrado o sistema que satisfaz:

$$\mathcal{LP} \hookrightarrow \text{“Provas reais”}$$

Estão ainda em aberto as seguintes questões:

- Estabelecer que $\mathcal{LP}$ realiza S4.
- Formalizar a semântica clássica BHK e demonstrar a completude de Int com respeito a esta semântica.

Pelos motivos já apresentados, iremos apresentar a sequência de resultados que nos permitiriam responder a estas questões, sem nos deter nas demonstrações. (cf. **Artemov, 1998**).

3.6.2 Definição

Dada uma fórmula F de $\mathcal{LP}$, F^0 é a fórmula resultante da substiuição de todas as ocorrências de $t : X$, em F, por $\Box X$.

Dado um conjunto Γ de fórmulas de $\mathcal{LP}$, $\Gamma^0 = \{F^0 | F \in \Gamma\}$.

3.6.3 Lema

Se $\mathcal{LP} \vdash F$ então S4 $\vdash F^0$.

Demonstração

Por indução na derivação de $\mathcal{LP}$.

Este lema fornece um processo para transformar cada derivação de $\mathcal{LP}$ numa derivação de S4. Resta verificar se $\mathcal{LP}$ é suficiente para realizar todos os teoremas de S4.

[39]Num modelo de **AP**, o elemento que satisfaz $\exists x P(x,F)$ pode ser não-standard. Nesse caso, $\exists x P(x,F)$ é verdadeira no modelo mas não existe uma derivação "real" em **AP** para esse x.

3.6.4 Definição

Uma **realização** r de uma fórmula modal F é uma aplicação que faz corresponder a cada ocorrência de $\Box$ em F, uma prova polinomial.

Designamos por F^r a imagem de F sob a realização r.

3.6.5 Definição (Polaridade positiva ou negativa de uma ocorrência de $\Box$)

Dizemos que uma ocorrência de $\Box$ em $\Box F$ tem polaridade positiva.

Uma ocorrência de $\Box$ antes de F nas fórmulas $G \to F$, $G \wedge F$, $F \wedge G$, $G \vee F$, $F \vee G$, $\Box F$ e $\Gamma \Rightarrow \Delta, F$ tem a mesma polaridade da ocorrência de $\Box$ **em** F.

Uma ocorrência de $\Box$ antes de F em $\neg F$, $F \to G$ e $F, \Gamma \Rightarrow \Delta$ tem a polaridade oposta da ocorrência de $\Box$ **em** F.

3.6.6 Definição

Uma realização r é **normal** se todas as ocorrências negativas de $\Box$ são realizadas por variáveis prova.

3.6.7 Teorema

Se S4 $\vdash F$ então $\mathcal{LP} \vdash F^r$, para alguma realização normal r.

Demonstração

Nesta demonstração é utilizada uma formulação de S4 através de sequências, sem a regra do corte (por forma a serem respeitadas as polaridades definidas).

Se S4 $\vdash F$ então existe uma derivação $\mathcal{T}$ do sequente $\Rightarrow F$. A demonstração passa por construir uma realização normal r tal que $\mathcal{LP} \vdash \bigwedge \Gamma^r \to \bigvee \Delta^r$ para cada sequência $\Gamma \Rightarrow \Delta$ em $\mathcal{T}$, fazendo corresponder a cada conectivo modal que ocorre na derivação de S4 uma prova polinomial de $\mathcal{LP}$. (cf. Artemov, 1998, demonstração do teorema 8.2, pág. 30)

Com o teorema anterior fechamos a cadeia *Int* $\hookrightarrow$ S4 $\hookrightarrow$ $\mathcal{LP}$ $\hookrightarrow$ "Provas reais".

3.6.8 Corolário (Completude aritmética de S4)

S4 $\vdash F$ sse existe uma realização r e uma especificação de constantes EC tais que F^r é EC-válida.

Conjugando os resultados obtidos no Lema 3.6.3 e no Teorema 3.6.7 concluímos que S4 não é mais do que uma versão "preguiçosa" de $\mathcal{LP}$ que não distingue os polinómios prova de cada fórmula.

3.6.9 Definição

Uma fórmula intuicionista proposicional F diz-se **GK-realizável** sse existe uma realização normal r tal que $gk(F)^r$ é derivável em $\mathcal{LP}$.

3.6.10 Teorema (Realização da lógica proposicional intuicionista)

Qualquer que seja a fórmula intuicionista F:

$Int \vdash F \;\; \Leftrightarrow F$ é GK-realizável, ou seja, $Int \vdash F \;\; \Leftrightarrow \mathcal{LP} \vdash gk(F)^r$, para alguma realização normal r.

Demonstração

Resulta da combinação do resultado $Int \vdash F \Leftrightarrow \mathrm{S4} \vdash gk(F)$ e da realização de S4 em $\mathcal{LP}$ (teorema 3.6.7).

3.6.11 Corolário (Completude aritmética de *Int*)

$Int \vdash F$ sse existe uma realização r e uma especificação de constantes EC tais que $gk(F)^r$ é EC-válida.

Considerações finais

Na lógica intuicionista, *criada* por Brouwer com base no raciocínio construtivista, o critério de verdade é a existência de uma prova. Contudo, na opinião de Brouwer esta prova não poderia ser encarada como uma dedução num qualquer sistema formal. Uma prova seria uma costrução mental intuitiva do matemático (ideal).

Apesar da aversão de Brouwer à formalização da matemática, Kolmogorov e Heyting sugeriram que a lógica intuicionista pudesse ser estudada formalmente, desenvolvendo uma interpretação informal (BHK) capaz de interiorizar os princípios da lógica intuicionista formulados por Brouwer.

Uma das aplicações desta semântica, que não tivemos oportunidade de referir e seria interessante explorar, é a capacidade de BHK introduzir uma "linguagem de programação". Baseados nesta semântica alguns autores, como Mints, encontraram um método de extrair programas a partir de provas intuicionistas, computando o termo associado à prova.

Os variadíssimos estudos que tiveram origem na procura da semântica adequada e desejada para BHK, resultaram no aparecimento de vários modelos de diferentes naturezas para a lógica intuicionista. Vimos como a semântica de Kripke, com a usual valuação (0-1) para as fórmulas atómicas e uma relação binária de acessibilidade, reflexiva e transitiva, se revela adequada à lógica intuicionista.

Relembremos que em cada estádio de conhecimento a validade da conjunção e da disjunção pode ser determinada de acordo com as tabelas de verdade clássicas. A implicação e a negação são verdadeiras num estádio sse forem verdadeiras classicamente em cada estádio acessível a partir dele.

É de lamentar, no entanto, não ser possível evitar um raciocínio clássico na completude da lógica de predicados intuicionista com respeito a esta semântica.

Nesta dissertação vimos ainda como a solução de Artemov para a questão da formalização de BHK, envolve interpretar *Int* numa lógica $\mathcal{LP}$ de proposições e provas. Os objectos prova são representados, no sistema $\mathcal{LP}$, por polinómios prova que podem ser concatenados e aplicados uns aos outros. $\mathcal{LP}$ acomoda não só a operação "aplicação" mas também o "verificador de provas" e a operação "escolha".

Embora as variáveis e conectivos proposicionais sejam análogos aos clássicos, Artemov introduz em $\mathcal{LP}$ uma nova proposição $t : \varphi$ para denotar que t é uma prova de φ. Para que $\mathcal{LP}$ englobe também a interpretação do axioma $\Box F \rightarrow \Box\Box F$ de S4, Artemov introduz o operador adicional ! para que $!t$ denote uma prova de que t é a prova de φ. Assim, para qualquer termo t e sentença φ, $\mathcal{LP}$ contém um axioma da forma $t : \varphi \rightarrow\ !t : t : \varphi$.

Na secção 3.5 vimos como encontrar uma interpretação aritmética de $\mathcal{LP}$, através de um predicado prova, em **AP**, e provámos a validade e completude de $\mathcal{LP}$ com respeito a essa interpretação. A título de exemplo de uma aplicação deste resultado, refira-se que a completude de $\mathcal{LP}$ permite obter a eliminação da regra do corte numa determinada formulação de $\mathcal{LP}$ através de sequências (cf. Artemov, 1998, corolário 7.12, pág. 28), no entanto, a mesma propriedade é obtida noutros sistemas (Troelstra 1988).

Na última secção referimos que $\mathcal{LP}$ cria um ambiente em que os operadores modais podem ser considerados provas polinomiais (teorema 3.6.7). O facto de uma derivação de $\mathcal{LP}$ ser projectada numa derivação de S4 e, reciprocamente, uma fórmula derivável em S4 ter uma realização derivável em $\mathcal{LP}$, é a chave para obter a completude da lógica proposicional intuicionista.

Apêndice

Semânticas

Interpretação BHK (pág. 14)

1. A é uma demonstração de $\varphi \wedge \psi$ sse A é um par ordenado (B, C) tal que B é uma demonstração de φ e C é uma demonstração de ψ.

2. A é uma demonstração de $\varphi \vee \psi$ sse A é um par ordenado (a, C), em que $a \in \{0, 1\}$ e C é uma construção, tal que, se $a = 0$ então C demonstra φ, e se $a = 1$ então C demonstra ψ.

3. A é uma demonstração de $\varphi \rightarrow \psi$ sse A é uma construção que converte cada demonstração B de φ numa demonstração $A(B)$ de ψ.

4. Nenhuma construção é uma demonstração de $\perp$.

5. A é uma demonstração de $\forall x \varphi(x)$ sse A é uma construção tal que, para cada $d \in \mathbf{D}$, $A(d)$ é uma demonstração de $\varphi(d)$.

6. A é uma demonstração de $\exists x \varphi(x)$ sse A é um par ordenado (d, C) tal que $d \in \mathbf{D}$ e C é uma demonstração de $\varphi(d)$.

Modelo de Kripke (pág. 28)

Quádruplo $\mathfrak{K} \equiv (K, \leq, D, \vDash)$, com K não vazio, tal que:

(i) $(K, \leq)$ é um conjunto parcialmente ordenado.

(ii) D é uma função (*função domínio*) que a cada elemento, k, de K, faz corresponder um conjunto não vazio $D(k)$, tal que: Quaisquer que sejam k e k' pertencentes a K, se $k \leq k'$ então $D(k) \subseteq D(k')$. Ou seja, D é monótona crescente (em sentido lato). Ao conjunto $D(k)$ chamamos domínio de $\mathfrak{K}$ no estádio k.

(iii) $\vDash$ é uma relação binária em $K \times \mathcal{P}$, sendo $\mathcal{P}$ o conjunto das fórmulas atómicas, tal que: Se $k \vDash P$ e $k' \geq k$ então $k' \vDash P$.

Extensão de $\vDash$ às fórmulas compostas:

$\mathbf{K_1}$. Não existe nenhum elemento $k \in K$ tal que $k \vDash \perp$.

$\mathbf{K_2}$. $k \vDash \varphi \wedge \psi := k \vDash \varphi$ e $k \vDash \psi$

$\mathbf{K_3}$. $k \vDash \varphi \vee \psi := k \vDash \varphi$ ou $k \vDash \psi$

$\mathbf{K_4}$. $k \vDash \varphi \rightarrow \psi :=$ Para todo o k' pertencente a K, se $k' \geq k$ e $k' \vDash \varphi$ então $k' \vDash \psi$.

$\mathbf{K_5}$. $k \vDash \forall x \varphi(x) := \forall k' \geq k \forall d \in D(k')\big(k' \vDash \varphi(d)\big)$

$\mathbf{K_6}$. $k \vDash \exists x \varphi(x) := \exists d \in D(k)\big(k \vDash \varphi(d)\big)$

Sistemas

Sistema de dedução natural (pág. 19)

Regras de Introdução	Regras de Eliminação
$(\wedge^+)\ \dfrac{\varphi\ \psi}{\varphi\wedge\psi}$	$(\wedge_1^-)\ \dfrac{\varphi\wedge\psi}{\varphi}\ (\wedge_2^-)\ \dfrac{\varphi\wedge\psi}{\psi}$
$(\vee_1^+)\ \dfrac{\varphi}{\varphi\vee\psi}\ (\vee_2^+)\ \dfrac{\psi}{\varphi\vee\psi}$	$(\vee^-)\ \dfrac{\varphi\vee\psi \quad \begin{matrix}[\varphi]\\ \mathcal{D}_1\\ \sigma\end{matrix} \quad \begin{matrix}[\psi]\\ \mathcal{D}_2\\ \sigma\end{matrix}}{\sigma}$
$(\rightarrow^+)\ \dfrac{\begin{matrix}[\varphi]\\ \mathcal{D}\\ \psi\end{matrix}}{\varphi\rightarrow\psi}$	$(\rightarrow^-)\ \dfrac{\varphi\ \ \varphi\rightarrow\psi}{\psi}$ (MP)
	$(\perp)\ \dfrac{\perp}{\varphi}$
$(\forall^+)\ \dfrac{\varphi(\alpha)}{\forall x\varphi(x)}$	$(\forall^-)\ \dfrac{\forall x\varphi(x)}{\varphi(t)}$
$(\exists^+)\ \dfrac{\varphi(t)}{\exists x\varphi(x)}$	$(\exists^-)\ \dfrac{\exists x\varphi(x) \quad \begin{matrix}[\varphi(a_0)]\\ \mathcal{D}\\ \sigma\end{matrix}}{\sigma}$

Sistema $\mathcal{LP}$ (pág. 45)

Axiomas:

A_0) Número finito de axiomas-esquema da lógica clássica proposicional, na linguagem de $\mathcal{LP}$.

A_1) $t : F \rightarrow F$ (Axioma de **verificação**)

A_2) $t : (F \rightarrow G) \rightarrow (s : F \rightarrow (t.s) : G)$ (Axioma de **aplicação**)

A_3) $t : F \rightarrow !t : (t : F)$ (Axioma do **verificador de provas**)

A_4) $t : F \rightarrow (s+t) : F,\ s : F \rightarrow (s+t) : F$ (Axioma de **escolha**)

Regras de inferência:

R_1) $\dfrac{F\rightarrow G\ \ F}{G}$ **(Modus Ponens)**

R_2) $\dfrac{}{c:A}$ sendo A um axioma do tipo A_0-A_4 **(Necessidade de Axioma)**

Para obter o **sistema $\mathcal{LP}_0$**, retiramos ao sistema $\mathcal{LP}$ a regra R_2.

Sistema $\mathcal{LPG}$ (pág. 62)

Axiomas:

Sequências do tipo: $\Gamma,\ F \Rightarrow F, \Delta$ ou $\Gamma,\ \perp\ \Rightarrow \Delta$

Regras:

Regra do corte: $\dfrac{\Gamma_1 \Rightarrow \Delta_1, A \quad A, \Gamma_2 \Rightarrow \Delta_2}{\Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2}$

Regras de introdução dos operadores lógicos:

$$\frac{A,\Gamma\Rightarrow\Delta}{A\wedge B,\Gamma\Rightarrow\Delta} \text{ ou } \frac{B,\Gamma\Rightarrow\Delta}{A\wedge B,\Gamma\Rightarrow\Delta}\ (\mathrm{L}\wedge) \qquad \frac{\Gamma\Rightarrow\Delta,A \quad \Gamma\Rightarrow\Delta,B}{\Gamma\Rightarrow\Delta,A\wedge B}\ (\mathrm{R}\wedge)$$

$$\frac{A,\Gamma\Rightarrow\Delta \quad B,\Gamma\Rightarrow\Delta}{A\vee B,\Gamma\Rightarrow\Delta}\ (\mathrm{L}\vee) \qquad \frac{\Gamma\Rightarrow\Delta,A}{\Gamma\Rightarrow\Delta,A\vee B} \text{ ou } \frac{\Gamma\Rightarrow\Delta,B}{\Gamma\Rightarrow\Delta,A\vee B}\ (\mathrm{R}\vee)$$

$$\frac{\Gamma\Rightarrow\Delta,A \quad B,\Gamma\Rightarrow\Delta}{A\to B,\Gamma\Rightarrow\Delta}\ (\mathrm{L}\to) \qquad \frac{A,\Gamma\Rightarrow\Delta,B}{\Gamma\Rightarrow\Delta,A\to B}\ (\mathrm{R}\to)$$

Regras estrutural de contracção:

$$\frac{A,A,\Gamma\Rightarrow\Delta}{A,\Gamma\Rightarrow\Delta}\ (\mathrm{LC}) \qquad \frac{\Gamma\Rightarrow\Delta,A,A}{\Gamma\Rightarrow\Delta,A}\ (\mathrm{RC})$$

E ainda as seguintes regras:

$$\frac{A,\Gamma\Rightarrow\Delta}{t{:}A,\Gamma\Rightarrow\Delta}\ (:\ \Rightarrow) \qquad \frac{\Gamma\Rightarrow\Delta,t{:}A}{\Gamma\Rightarrow\Delta,!t{:}t{:}A}\ (\Rightarrow\ !)$$

$$\frac{\Gamma\Rightarrow\Delta,t{:}A}{\Gamma\Rightarrow\Delta,(t+s){:}A}\ (\Rightarrow\ +) \qquad \frac{\Gamma\Rightarrow\Delta,t{:}A}{\Gamma\Rightarrow\Delta,(s+t){:}A}\ (\Rightarrow\ +)$$

$$\frac{\Gamma\Rightarrow\Delta,t{:}(A\to B) \quad \Gamma\Rightarrow\Delta,s{:}A}{\Gamma\Rightarrow\Delta,(ts){:}B}\ (\Rightarrow\ .)$$

$$\frac{}{\Gamma\Rightarrow c{:}A,\Delta}\ (\Rightarrow c).$$

Para obter o **sistema $\mathcal{LPG}_0$**, retiramos ao sistema $\mathcal{LPG}$ a regra ($\Rightarrow c$).

Retirando a regra do corte a $\mathcal{LPG}$ e $\mathcal{LPG}_0$, obtemos os **sistemas $\mathcal{LPG}^-$** e **$\mathcal{LPG}_0^-$**, respectivamente.

Aritmética de Peano AP (pág. 51)

Axiomas $AP_1 - AP_{10}$ e axioma de indução:

AP_1. $\forall x(x' \neq 0)$

AP_2. $\forall xy(x' = y' \rightarrow x = y)$

AP_3. $\forall x(x + 0 = x)$

AP_4. $\forall xy\big(x + y' = (x + y)'\big)$

AP_5. $\forall x(x \times 0 = 0)$

AP_6. $\forall x\big(x \times y' = x \times y + x\big)$

AP_7. $\forall x\big(x \uparrow 0 = 0'\big)$

AP_8. $\forall xy\big(x \uparrow y' = x \uparrow y \times x\big)$

Axiomas de ordem:

AP_9. $\forall x\big(x \not< 0\big)$

AP_{10}. $\forall xy\big(x < y' \leftrightarrow x \leq y\big)$

Axioma de indução:

Axioma-esquema (Ind) $\forall \overline{y}\Big(\varphi(0, \overline{y}) \wedge \forall x\big(\varphi(x, \overline{y}) \rightarrow \varphi(x', \overline{y})\big) \rightarrow \forall x \varphi(x, \overline{y})\Big)$

Referências bibliográficas

[1] S. Artemov, "Explicit provability: the intended semantics for intuitionistic and modal logic", *Technical Report CFIS 98-10*, Cornell University, September 1998.

[2] S. Artemov, "Kolmogorov and Gödel's approach to intuitionistic logic: current developments", *Russian Math. Surveys* 59, n.º2, pp. 203-229, 2004.

[3] S. Artemov, "Logic of Proofs", *Annals of Pure and Applied Logic*, v.67, pp. 29-59, 1994.

[4] S. Artemov, "Logic of Proofs: a Unified Semantics for Modality and λ-terms", *Technical Report CFIS 98-06*, Cornell University, March 1998.

[5] S. Artemov, "Operational Modal Logic", *Technical Report MSI 95-29*, Cornell University, December 1995.

[6] S. Artemov, "Understanding Constructive Semantics", *Publicação interna*, Cornell and Moscow University, August 1999.

[7] J. Barwise, *Handbook of Mathematical Logic*, Eight Edition, North Holland, 1993.

[8] G. Boolos, *The Logic of Provability*, Cambridge University Press, 1993.

[9] D. van Dalen, *Logic and Structure*, Second Edition, Springer-Verlag 1983.

[10] M. Dummet, *Elements of Intuitionism*, Second Edition, Oxford, Clarendon Press, 2000.

[11] H. B. Enderton, *A Mathematical Introduction to Logic*, Second Edition, University of California, Los Angeles, Harcourt Academic Press, 2001.

[12] R. L. Epstein, *The semantic foundations of logic. Proposicional logics*, Second Edition, Oxford University Press, New York, 1995.

[13] W. Felscher, Lectures on Mathematical Logic Volume III: The Logic of Arithmetic, Gordon and Breach Science Publishers, 2000.

[14] E. Mendelson, *Introduction to Mathematical Logic*, Fourth Edition, Chapman & Hall, 1997.

[15] G. Mints, *A Short Introduction to Intutionistic Logic*, Kluwer Academic/Plenum Publishers, 2000.

[16] A. J. F. Oliveira, *Lógica e Aritmética*, 2.ª edição, Gradiva, 1996.

[17] H. Rogers, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, 1967.

[18] A. S. Troelstra and D. van Dalen, *Constructivism in Mathematics. An Introduction*, v. 1, Amsterdam; North Holland, 1988.

[19] A. S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*, Second Edition, Cambridge University Press, 2000.