

2

Conceitos Fundamentais sobre Prova de Teoremas

Este capítulo apresenta uma revisão dos conceitos fundamentais sobre prova de teoremas. Na primeira parte deste capítulo, definimos os conceitos de lógica, tratamos da distinção entre semântica e sintaxe, apresentamos a teoria sobre sistemas dedutivos e introduzimos alguns aspectos fundamentais sobre teoria da prova. A parte final do capítulo explora o conceito de prova automática e prova interativa, introduzindo os provadores de teoremas automatizados. As referências básicas para os conceitos aqui apresentados são (End72), (MH06) e (SFM06). Referências específicas são indicadas ao longo do texto.

2.1

Linguagens Lógicas

Lógica pode ser vista como o estudo da argumentação. Mais especificamente, é o estudo de como avaliar os argumentos, distinguindo entre aqueles que são válidos e os que não são. Uma *argumentação* é uma sequência de sentenças onde cada sentença é obtida a partir das anteriores através de uma inferência até a obtenção de uma conclusão final. Na Lógica Matemática, uma argumentação também é conhecida como *prova* ou *demonstração*. Cada passo onde uma nova sentença (*conclusão*) é derivada de um conjunto de sentenças anteriores (*premissas*), corresponde à aplicação de uma *regra de inferência* ou *argumento básico*. Dessa forma, uma prova corresponde a uma sequência de aplicações de regras de inferência. Considerando que regras de inferência podem ter mais de uma premissa e apenas uma conclusão, uma prova pode ser adequadamente representada como uma *árvore*. Todas estas definições são estritamente sintáticas, isto é, estamos falando basicamente da manipulação de símbolos sem considerar seus correspondentes significados.

Uma *lógica*, mais especificamente, uma *linguagem lógica* é uma linguagem usada para escrever sentenças. A semântica de uma lógica é dada pelos possíveis valores que expressões escritas nesta lógica podem assumir. Para linguagens baseadas na Lógica Clássica, como a lógica proposicional e a lógica de primeira ordem, os valores permitidos para as sentenças podem ser

“verdadeiro” ou “falso”. Lógicas enfocando outros tipos de juízos existem. Por exemplo, na Lógica Temporal, o juízo atribuído a uma sentença leva em consideração a noção de tempo. Além disso, em algumas lógicas não clássicas, denominadas Lógicas Multivaloradas, sentenças podem assumir mais do que apenas estes dois valores. Um caso particular é o da Lógica Ternária, ou Lógica de Três Valores, onde sentenças podem ser consideradas “verdadeiras”, “falsas” ou “indefinidas”.

As sentenças que podem ser representadas por uma linguagem são expressas por meio de *fórmulas bem formadas*, ou simplesmente, fórmulas, isto é, cadeias de caracteres que obedecem às regras sintáticas (gramática) da linguagem. Fórmulas são escritas usando símbolos lógicos (conectivos, quantificadores, parênteses, variáveis e o símbolo \perp) e não lógicos (letras proposicionais, símbolos predicativos, funcionais e constantes). Uma linguagem lógica estabelece quais símbolos lógicos e não lógicos podem ser usados para escrever fórmulas nesta linguagem.

O que pode ser expresso varia de linguagem para linguagem. Chamamos a capacidade de uma linguagem lógica de representar sentenças de *expressividade*. Uma lógica pode ser capaz de representar um determinado problema enquanto outra não. Quanto maior a expressividade, maior também a complexidade de se manipular uma linguagem. Abaixo apresentamos duas linguagens fundamentais da Lógica Clássica, em ordem crescente de acordo com sua expressividade.

A *lógica proposicional* é uma linguagem bastante utilizada na Lógica Clássica que permite estabelecer relações lógicas sobre *proposições*. Uma proposição equivale a uma sentença em linguagem natural a qual podemos atribuir os valores verdadeiro ou falso, como por exemplo:

Está chovendo.

Todo homem é mortal.

Na lógica proposicional, sentenças como as acima são codificadas usando letras proposicionais que podem assumir valores booleanos. A linguagem da lógica proposicional inclui as letras proposicionais, os conectivos booleanos tradicionais e os parênteses. Podemos especificar formalmente a linguagem através de uma gramática livre de contexto em Forma de Backus-Naur (notação BNF), como mostrado abaixo:

$$\phi ::= p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi)$$

onde p é usado genericamente, podendo ser qualquer letra, representando uma proposição

A linguagem proposicional tem uma expressividade bastante limitada. Ela não nos permite expressar, por exemplo, relações sobre elementos de um conjunto, como as noções de “todos”, “algum” ou “nenhum”. Como exemplo, a sentença abaixo não pode ser expressada usando a lógica proposicional:

$$\text{Para todos os números naturais } x, y : x + y = y + x \quad (2-1)$$

Para resolver o problema de expressividade da lógica proposicional foi criada a lógica de predicados, ou *lógica de primeira ordem*. Adicionalmente aos símbolos da lógica proposicional, a lógica de primeira ordem inclui os quantificadores “para todo” e “existe”, os símbolos predicativos ou funcionais, variáveis e constantes. Símbolos predicativos ou funcionais são, respectivamente, relações ou funções sobre um domínio arbitrário. A gramática livre de contexto em BNF para a lógica de primeira ordem é como segue:

$$\phi ::= P(t_1, t_2, \dots, t_n) \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid \forall x\phi \mid \exists x\phi$$

onde x é usado genericamente, podendo ser qualquer letra, representando uma variável

P é um predicado ou função com n argumentos, $n \in \mathbb{N}$. O número total de argumentos de um predicado ou função recebe o nome de *aridade*. Na lógica de primeira ordem, os componentes de uma fórmula que nomeiam objetos, são chamados *termos*. Portanto, cada t_i é um termo. x é uma variável. Assim, considerando o conjunto dos números naturais \mathbb{N} como o domínio para variáveis e definindo a função “+” e o predicado “=”, ambos binários, isto é, ambos de aridade 2, podemos representar a sentença da equação (2-1), como:

$$\forall x, y (= (+(x, y), +(y, x)))$$

onde $\forall x, y$ é uma abreviação para $\forall x\forall y$.

Ou ainda, usando notação infixa:

$$\forall x, y (x + y = y + x)$$

Termos são construídos a partir das constantes, variáveis e funções da linguagem. *Fórmulas atômicas* são fórmulas que não podem ser decompostas em outras fórmulas. Fórmulas atômicas são fórmulas bem formadas que não contém nenhum conectivo ou quantificador e são construídas a partir dos predicados da linguagem. Por exemplo, a fórmula $< (0, +(1, 2))$ é uma fórmula atômica, onde 0 , $+(1, 2)$, 1 e 2 são termos. Já a fórmula da equação (2.1) não é atômica.

Como os exemplos acima, podemos ver claramente como a lógica de primeira ordem é mais poderosa que a proposicional. Na verdade, a lógica de primeira ordem é a linguagem fundamental da matemática e com ela foram formuladas, por exemplo, a teoria dos números e a geometria. Mas apesar disso tudo, ela também tem suas limitações quanto a capacidade de expressar sentenças. Isto ocorre pelo fato de que, em primeira ordem, os quantificadores e predicados variam apenas sobre variáveis, o que impede produzir sentenças que, por exemplo, falem sobre predicados. O conceito matemático de indução é um exemplo de sentença em que precisamos de um pouco mais de expressividade do que a oferecida pela lógica de primeira ordem.

$$\forall P ((P(0) \wedge \forall n (P(n) \rightarrow P(n+1))) \rightarrow \forall n P(n))$$

A sentença acima afirma que para cada predicado P , se $P(0)$ vale e se, para todo $n \in \mathbb{N}$, $P(n)$ também vale e podemos concluir, a partir dessa afirmação, que vale também $P(n+1)$, conclui-se que, $P(n)$ vale para todo n . Tal afirmação corresponde ao que determina a indução matemática. Claramente, não podemos escrever tal sentença com a lógica de primeira ordem, porque ela não quantifica sobre predicados. Para este tipo de afirmação usamos outra linguagem, a lógica de segunda ordem. Esta última, por sua vez, pertence a classe das *lógicas de mais alta ordem* (em inglês, *higher order logics*). Provadores de teoremas como HOL (Hol) são baseados nesse tipo de linguagem.

2.2

Semântica

Para sermos capazes de raciocinar sobre a verdade de uma sentença escrita em uma linguagem lógica, precisamos de algum mecanismo para atribuir significado a essas sentenças. Quando escrevemos fórmulas numa linguagem lógica, os símbolos lógicos permitidos seguem seu significado intrínseco, mas é preciso definir uma forma para interpretar os símbolos não lógicos. Chamamos de *interpretação* à atribuição de significado aos símbolos não lógicos de uma linguagem lógica. Na lógica proposicional isto é feito atribuindo os valores “verdadeiro” ou “falso” às letras proposicionais em uma fórmula, o que corresponde a uma linha da tabela verdade da sentença. Para a lógica de primeira ordem, o significado de uma fórmula depende do estabelecimento de uma *estrutura*. Uma estrutura determina:

1. Um conjunto \mathfrak{D} de indivíduos sobre o qual variáveis e constantes assumem valores, chamado de universo de discurso.

2. Para cada símbolo predicativo, uma relação $R \subseteq \mathfrak{D}^n$, ou seja, R é um conjunto de n-tuplas de membros de \mathfrak{D} .
3. Para cada símbolo constante, um elemento de \mathfrak{D} .
4. Para cada símbolo funcional, uma função $f : \mathfrak{D}^n \rightarrow \mathfrak{D}$

Por exemplo, a estrutura $\langle \mathbb{Z}, +, * \rangle$ representa a interpretação usada para expressar a Teoria dos Números.

Dizemos que uma fórmula é *válida* quando a mesma é verdadeira em todas as interpretações. Para a lógica proposicional uma fórmula válida é chamada de *tautologia* e, neste caso, corresponde às fórmulas cuja tabela verdade tem a última coluna sempre verdadeira. Na lógica de primeira ordem, fórmulas válidas são aquelas verdadeiras em todas as estruturas possíveis.

Uma fórmula é *válida em uma interpretação* se é sempre verdadeira naquela interpretação. Na lógica proposicional isto equivale a uma linha de uma tabela verdade que termina em verdadeiro. Em primeira ordem isto equivale a uma fórmula que é verdade na estrutura em questão.

Uma fórmula ϕ é dita ser *satisfatível* se existe alguma interpretação onde ϕ é verdade.

Uma fórmula ϕ é dita ser *falsificável* se existe alguma interpretação onde ϕ é falsa.

Por fim, uma fórmula ϕ é dita ser *insatisfatível* se ϕ é sempre falsa para qualquer interpretação dada. Se ϕ é insatisfatível, $\neg\phi$ é válida e vice-versa.

O conceito de *consequência lógica* estabelece uma relação de decorrência entre um conjunto de fórmulas e uma fórmula. Isto é, a consequência lógica determina quando uma fórmula decorre logicamente de um conjunto de fórmulas. Formalmente, sendo Δ um conjunto de fórmulas e α uma fórmula, temos:

$$\Delta \models \alpha$$

se e somente se toda interpretação que satisfaz todas as fórmulas de Δ também satisfaz α . Dessa forma, uma fórmula válida α é representada como $\models \alpha$.

O conceito de consequência é central em Lógica. Dissemos no início deste capítulo que cada passo do processo de prova corresponde à aplicação de uma regra de inferência. Pela consequência lógica, temos que uma regra de inferência é correta quando, sob qualquer possível atribuição de valores verdade, se as premissas são verdadeiras, então a conclusão também é verdadeira.

2.3

Sistemas Dedutivos

A consequência lógica é um conceito semântico, isto é, depende do significado das fórmulas em questão. Um *sistema dedutivo* é um mecanismo sintático correspondente à consequência lógica, ou seja, sistemas dedutivos também servem para analisar a validade de fórmulas, mas focando exclusivamente na representação simbólica das mesmas.

Um sistema ou cálculo dedutivo corresponde a um conjunto de *axiomas* (possivelmente vazio) e um conjunto de *regras de inferências* usadas para raciocinar sobre fórmulas. Regras de inferência já foram discutidas nas seções anteriores. Axiomas correspondem a fórmulas que não necessitam de prova, isto é, um axioma é uma sentença considerada óbvia ou aceita como um consenso inicial necessário para outras deduções no sistema dedutivo.

Para dizer que uma fórmula decorre logicamente de um conjunto de fórmulas no nível sintático usamos:

$$\Delta \vdash \alpha$$

O conjunto de fórmulas Δ é também chamado de *conjunto de hipóteses*. Quando conseguimos concluir que uma determinada fórmula decorre de um determinado conjunto de hipóteses, temos um *teorema*. Podemos ver um sistema dedutivo como um mecanismo para provar teoremas. Neste caso, uma prova é uma sequência finita de passos de inferência onde uma conclusão é obtida usando somente regras de inferência e axiomas. Uma prova, portanto, pode ser vista como um artefato sintático, que envolve simplesmente manipulação simbólica. Em outras palavras, uma fórmula é um teorema se pode ser provada usando algum sistema dedutivo. Axiomas são considerados teoremas, já que são aceitos como verdadeiros no sistema dedutivo em questão.

Existe uma forte relação entre a noção semântica (verdade de fórmulas) e a noção sintática (provar fórmulas) quando analisamos o fato de uma fórmula decorrer logicamente de um conjunto de fórmulas. As definições de completude e correção esclarecem esta relação, estabelecendo a equivalência entre as duas visões:

- Correção: expressa que se $\Delta \vdash \alpha \Rightarrow \Delta \models \alpha$.
- Completude: expressa que se $\Delta \models \alpha \Rightarrow \Delta \vdash \alpha$.

Se uma prova conclui α sem utilizar hipóteses, temos o equivalente sintático ao conceito de fórmula válida. Representamos este caso por $\vdash \alpha$. Neste tipo de prova, se considerarmos que o sistema dedutivo usado para realizar a prova é correto e completo temos:

- pela correção do sistema dedutivo, um teorema é uma fórmula válida.
- pela completude, toda fórmula válida é um teorema que possui uma prova sem hipóteses.

A formalização de sistemas dedutivos como conhecemos hoje, tem sua origem no final do século XIX, com o famoso trabalho de Frege conhecido como *Begriffsschrift*, o primeiro esforço abrangente de desenvolver uma linguagem formal capaz de servir como fundamento para a matemática. Pouco tempo depois, Russel descobriu um paradoxo que mostrava que o sistema de Frege era inconsistente, isto é, a verdade de qualquer proposição podia ser provada nele. Russel, então, criou um sistema próprio, usando a teoria dos tipos como solução para seu paradoxo e, ele e Whitehead, demonstraram como este sistema poderia servir como base para a matemática, no célebre trabalho denominado *Principia Mathematica*. Em 1900, Hilbert enunciou uma lista com 23 problemas em aberto da matemática no início daquele século. Provar a consistência da aritmética, isto é, a impossibilidade de se provar o absurdo nela, era um desses problemas. Tentando resolvê-lo, Hilbert aperfeiçoou o sistema de Frege, criando um sistema dedutivo com axiomas e regras de inferência. Esse sistema serviu de modelo para muitos outros, que passaram a ser conhecidos como *sistemas dedutivos a la Hilbert*.

Em 1935, Gödel provou a não provabilidade da consistência da aritmética, isto é, ele mostrou que não é possível provar a consistência da aritmética usando apenas elementos da própria aritmética, no que ficou conhecido como *Teorema da Incompletude*. Em 1936, Gentzen, trabalhando na prova da consistência, estendeu o trabalho de Hilbert, criando o sistema dedutivo que ficou conhecido como *dedução natural*, no qual o significado de cada conectivo lógico é explicado através de um conjunto de regras de inferência. O teorema conhecido como *Hauptsatz* foi a base de sua prova da consistência. Como um dispositivo técnico para auxiliar nessa prova, ele extraiu de seu sistema de dedução natural o *cálculo de sequentes* e mostrou que este deriva os mesmos teoremas que o primeiro. O sistema de dedução natural foi aperfeiçoado, décadas mais tarde, em 1965, por Prawitz. Pelletier apresenta as bases históricas do sistema de dedução natural em (Pel00), enquanto Mackenzie narra, em detalhes, a história da lógica moderna em (Mac95).

Os resultados alcançados a partir do trabalho destes pioneiros consiste no objeto de estudo da área da Lógica denominada *Teoria da Prova*. Troelstra e Schwichtenberg apresentam em detalhes estes resultados em (TS00). Nas subseções a seguir apresentamos um pequeno resumo do sistema dedutivo de dedução natural e do cálculo de sequentes de Gentzen.

2.4

Dedução Natural

Como dito acima, o sistema de dedução natural possui um conjunto de regras para cada conectivo lógico e, considerando a lógica de primeira ordem, para cada quantificador. O sistema não possui axiomas, apenas regras de inferência. As regras são divididas em *regras de introdução* e *regras de eliminação*, para cada conectivo e quantificador. Regras de introdução tem por conclusão uma fórmula com o conectivo ou quantificador como símbolo principal. Regras de eliminação tem uma fórmula com o conectivo ou quantificador na *premissa maior* da mesma. A premissa maior de uma regra funciona como sendo a que determina a regra. Considerando a regra de eliminação do conectivo \rightarrow , como exemplo, temos:

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} (\rightarrow -E)$$

onde a premissa maior é $\varphi \rightarrow \psi$, a conclusão ψ e a outra premissa φ é dita ser *premissa menor* da regra.

Um conceito central do sistema de dedução natural é o *mecanismo de descarte de hipóteses*, que é o processo pelo qual podemos desconsiderar, a partir de um ponto em uma demonstração, em função da conclusão obtida, a dependência de uma hipótese. O descarte de uma fórmula em uma dedução é denotada envolvendo essa fórmula com colchetes, como pode ser visto na regra de introdução do conectivo \rightarrow , abaixo.

$$\frac{\begin{array}{c} [\varphi] \\ \vdots \\ \psi \end{array}}{\varphi \rightarrow \psi} (\rightarrow -I)$$

significando que supomos φ e, em algum ponto depois, deduzimos ψ , a partir da suposição que fizemos de φ . Neste momento ocorre o descarte da hipótese, que não precisa desconsiderar todas as ocorrências de uma hipótese. Entretanto, para ser uma prova, uma dedução não pode possuir hipóteses, ou seja, todas as hipóteses, se existiam, foram descartadas.

Na tabela 2.1 apresentamos a lista completa das regras de introdução e eliminação dos conectivos e quantificadores da dedução natural, considerando:

- na regra \forall -Introdução, o parâmetro a não pode ocorrer em nenhuma hipótese da qual a prova de $A(a)$ dependa;
- na regra \exists -Eliminação, o parâmetro a não pode ocorrer em $\exists xA(x)$, em C ou em qualquer hipótese da qual a prova da ocorrência superior de C

dependa que não as descartadas pela regra.

$\frac{A \quad B}{A \wedge B} (\wedge - I)$	$\frac{A \wedge B}{A} (\wedge - E_{esq})$	$\frac{A \wedge B}{B} (\wedge - E_{dir})$
$\frac{A}{A \vee B} (\vee - I_{dir})$	$\frac{B}{A \vee B} (\vee - I_{esq})$	$\frac{A \vee B \quad \begin{array}{c} [A] \\ \vdots \\ C \end{array} \quad \begin{array}{c} [B] \\ \vdots \\ C \end{array}}{C} (\vee - E)$
$\frac{\begin{array}{c} [A] \\ \vdots \\ B \end{array}}{A \rightarrow B} (\rightarrow - I)$	$\frac{A \quad A \rightarrow B}{B} (\rightarrow - E)$	
$\frac{\perp}{\neg A} (\neg - I)$	$\frac{A \quad \neg A}{\perp} (\neg - E)$	
$\frac{A(t)}{\exists x A(x)} (\exists - I)$	$\frac{\exists x A(x) \quad \begin{array}{c} [A(a)] \\ \vdots \\ C \end{array}}{C} (\exists - E)$	
$\frac{A(a)}{\forall x A(x)} (\forall - I)$	$\frac{\forall x A(x)}{A(t)} (\forall - E)$	
$\frac{\perp}{A} (\perp - Intuicionista)$	$\frac{\perp}{A} (\perp - Clássico)$	

Tabela 2.1: Dedução Natural para Primeira Ordem Clássica

O símbolo \perp representa o *sempre falso* ou *absurdo*. Vale notar que as regras de introdução e eliminação do \neg são casos particulares das respectivas regras do conectivo \rightarrow , considerando que $\neg A$ equivale a $A \rightarrow \perp$. As duas últimas regras, referentes ao absurdo, complementam as regras dos conectivos e quantificadores, fazendo da dedução natural um sistema completo para a Lógica Clássica. A regra do absurdo intuicionista reflete a noção que temos a partir do conceito de consequência lógica de que do sempre falso pode-se concluir qualquer proposição. A regra do absurdo clássico, por sua vez, representa o conceito conhecido como *redução ao absurdo* onde, para provar uma asserção A , supomos $\neg A$ e tentamos provar o absurdo. Um artifício bem comum em provas matemáticas. Esta regra é fundamental para a Lógica Clássica, pois é ela que nos permite provar que $A \vee \neg A$, o *princípio do terceiro excluído*, que garante que sentenças só podem assumir dois valores possíveis, verdadeiro ou falso. A dedução natural tem esse nome por buscar funcionar de forma semelhante a maneira como os matemáticos costumam desenvolver suas provas.

Quando todas as regras da tabela 2.1 podem ser usadas, exceto a regra do \perp – *Clássico*, temos a definição de uma outra lógica, denominada *Lógica Intuicionista*. Na Lógica Clássica, a semântica dos conectivos lógicos é dada por *funções booleanas*, isto é, funções com domínios e contradomínios baseados no conjunto $\{V, F\}$. Por este motivo, todos os conectivos e quantificadores podem ser obtidos a partir de um só conectivo e um dos quantificadores, por meio de composição de funções. Na Lógica Intuicionista, por outro lado, todos são necessários, não sendo possível interdefiní-los dessa maneira.

Como exemplo de uma prova em dedução natural, temos a prova de

$$\neg A \vee B \vdash A \rightarrow B \tag{2-2}$$

como segue:

$$\frac{\frac{\overline{A}^{(1)} \quad \overline{[\neg A]}^{(2)}}{\perp} \quad \frac{\perp}{A \rightarrow B}^{(1)} \quad \frac{\overline{[B]}^{(2)}}{A \rightarrow B}^{(1)}}{\neg A \vee B} \quad \frac{\overline{[B]}^{(2)}}{A \rightarrow B}^{(2)}}{A \rightarrow B}$$

para facilitar a leitura, usamos índices ao lado da linha equivalente a aplicação de uma regra que passou a desconsiderar determinada hipótese. Este mecanismo permite identificarmos que regra descarta qual ocorrência de qual hipótese da dedução.

2.5 Cálculo de Sequentes

O cálculo de sequentes é um sistema dedutivo formado por um único axioma, $\Delta, A \Rightarrow \Gamma, A$, onde Δ e Γ são sequências de fórmulas e A é uma fórmula. Além disso possui dois conjuntos de regras, as regras lógicas, que correspondem aos conectivos lógicos usuais e as *regras estruturais*. São 3 regras estruturais que determinam certas propriedades características desse sistema dedutivo, como segue:

- **Atenuação:** hipóteses podem não ser usadas;
- **Permutação:** a ordem das hipóteses é irrelevante;
- **Contração:** hipóteses podem ser usadas mais de uma vez.

O cálculo de sequentes é um sistema bastante mecanizável, o que contribui para o seu uso como base para muitos processos automatizados, atualmente.

Um *sequente* é uma fórmula do tipo $\mathcal{B}_1, \dots, \mathcal{B}_n \vdash \mathcal{A}_1, \dots, \mathcal{A}_m$. O lado esquerdo do símbolo \vdash é chamado de antecedente (correspondente a sequência

de hipóteses) e o lado direito é o conseqüente (ou conclusão). No antecedente, a vírgula é lida como conjunção, enquanto no conseqüente ela é lida como disjunção.

O método de prova usando o cálculo de seqüentes consiste em, primeiramente, aplicar uma regra apropriada do sistema ao objetivo inicial, isto é, a fórmula que representa o teorema que se deseja provar e, portanto a conclusão final da possível prova. Isto produz novas fórmulas como premissas da fórmula anterior. O processo é repetido, aplicando regras apropriadas a cada uma destas novas fórmulas obtidas, que se tornam, então, conclusões de novas fórmulas (suas premissas) geradas. Esta seqüência de aplicação de regras se repete até que as fórmulas produzidas sejam todas axiomas do sistema, confirmando a prova da fórmula inicial, ou não seja mais possível aplicar alguma regra. Neste caso, as últimas premissas produzidas representam contra-exemplos que mostram que a fórmula que está tentando ser provada não é verdadeira. Este procedimento sempre termina quando consideramos linguagens lógicas decidíveis. Veremos no próximo capítulo que nem toda linguagem lógica é decidível e, mesmo naquelas que são, mas sem a regra de eliminação do corte, não podemos garantir que o procedimento descrito acima alcance um dos seus estados finais.

Abaixo apresentamos as regras lógicas e estruturais do cálculo de seqüentes para a lógica de primeira ordem clássica.

Conjunção

$$\frac{\Gamma \Rightarrow \Delta, F \quad \Gamma \Rightarrow \Delta, G}{\Gamma \Rightarrow \Delta, F \wedge G} (\wedge, dir)$$

$$\frac{F, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} (\wedge, esq_1) \quad \frac{G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} (\wedge, esq_2)$$

Disjunção

$$\frac{\Gamma \Rightarrow \Delta, F}{\Gamma \Rightarrow \Delta, F \vee G} (\vee, dir_1) \quad \frac{\Gamma \Rightarrow \Delta, G}{\Gamma \Rightarrow \Delta, F \vee G} (\vee, dir_2)$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} (\vee, esq)$$

Implicação

$$\frac{F, \Gamma \Rightarrow \Delta, G}{\Gamma \Rightarrow \Delta, F \rightarrow G} (\rightarrow, dir) \quad \frac{\Gamma \Rightarrow \Delta, F \quad G, \Gamma' \Rightarrow \Delta'}{F \rightarrow G, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'} (\rightarrow, esq)$$

Negação

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg F} (\neg, dir) \quad \frac{\Gamma \Rightarrow \Delta, F}{\neg F, \Gamma \Rightarrow \Delta} \neg, esq$$

Quantificador Universal

$$\frac{\Gamma \Rightarrow \Delta, F(a)}{\Gamma \Rightarrow \Delta, \forall x F(x)} (\forall, dir) \quad \frac{F(t), \Gamma \Rightarrow \Delta}{\forall x F(x), \Gamma \Rightarrow \Delta} (\forall, esq)$$

Quantificador Existencial

$$\frac{\Gamma \Rightarrow \Delta, F(t)}{\Gamma \Rightarrow \Delta, \exists x F(x)} (\exists, dir) \quad \frac{F(a), \Gamma \Rightarrow \Delta}{\exists x F(x), \Gamma \Rightarrow \Delta} (\exists, esq)$$

Atenuação

$$\frac{\Gamma \Rightarrow \Delta}{F, \Gamma \Rightarrow \Delta} (W, esq) \quad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, F} (W, dir)$$

Contração

$$\frac{F, F, \Gamma \Rightarrow \Delta}{F, \Gamma \Rightarrow \Delta} (C, esq) \quad \frac{\Gamma \Rightarrow \Delta, F, F}{\Gamma \Rightarrow \Delta, F} (C, dir)$$

Permutação

$$\frac{\Pi, G, F, \Gamma \Rightarrow \Delta}{\Pi, F, G, \Gamma \Rightarrow \Delta} (P, esq) \quad \frac{\Gamma \Rightarrow \Delta, G, F, \Sigma}{\Gamma \Rightarrow \Delta, F, G, \Sigma} (P, dir)$$

Corte

$$\frac{\Gamma \Rightarrow \Delta, F \quad F, \Pi \Rightarrow \Sigma}{\Gamma, \Pi \Rightarrow \Delta, \Sigma} (cut)$$

A prova da fórmula (2-2) usando cálculo de seqüentes é como segue:

$$\frac{\frac{A \Rightarrow B, A}{\neg A, A \Rightarrow B} \quad B, A \Rightarrow B}{A, \neg A \vee B \Rightarrow B} \quad \frac{A, \neg A \vee B \Rightarrow B}{\neg A \vee B \Rightarrow A \rightarrow B}$$

Gentzen mostrou que para provas com um certo “padrão” específico, não é possível concluir o absurdo, pois, neste padrão, toda fórmula que ocorra na dedução ou é subfórmula da conclusão, ou de alguma das hipóteses. O *Teorema da Eliminação do Corte* ou *Hauptsatz*, no cálculo de seqüentes, e o da *Normalização*, na dedução natural, garantem que qualquer prova pode ser transformada para uma prova equivalente que segue este padrão, chamadas de *provas normais*, garantindo assim, a consistência do sistema.

Como dissemos na seção anterior, Gentzen criou o cálculo de seqüentes como um mecanismo auxiliar para analisar provas em dedução natural, mostrando que este sistema prova os mesmos teoremas que a dedução natural. Abaixo, mostramos a relação para as regras dos conectivos \wedge e \rightarrow entre os

dois sistemas, considerando a Lógica Intuicionista.

Correspondência entre as Regras do \wedge

$$\frac{\frac{\Gamma \quad \Gamma}{A \quad B} (\wedge - Intro)}{A \wedge B} \quad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B} (\wedge - dir)$$

$$\frac{\frac{A \wedge B}{A}, \Gamma}{C} (\wedge - Elim) \quad \frac{\Gamma, A \Rightarrow C}{\Gamma, A \rightarrow B \Rightarrow C} (\wedge - esq)$$

Correspondência entre as Regras do \rightarrow

$$\frac{\frac{\Gamma, [A]}{B}}{A \rightarrow B} (\rightarrow -Intro) \quad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B} (\rightarrow -dir)$$

$$\frac{\frac{\Gamma}{A} \quad A \rightarrow B}{B} (\rightarrow -Elim) \quad \frac{\Gamma \Rightarrow A \quad B, \Gamma \Rightarrow C}{\Gamma, A \rightarrow B \Rightarrow C} (\rightarrow -esq)$$

Este raciocínio pode ser levado para cada um dos demais conectivos lógicos, demonstrando a equivalência entre os dois sistemas. Com algumas variações é possível adaptá-los para a Lógica Clássica também. As regras estruturais evidenciam propriedades intrínsecas do sistema de dedução natural e são necessárias para garantir a equivalência entre os dois sistemas.

2.6

Tableaux

O método dos *tableaux* analíticos é um sistema dedutivo baseado em um procedimento de decisão para a lógica proposicional e de semi-decisão para a lógica de primeira ordem. Tal procedimento nos permite definir se uma fórmula é consequência lógica de um conjunto de fórmulas ou não. Ou seja, podemos determinar se $\mathcal{B}_1, \dots, \mathcal{B}_n \vdash \mathcal{A}_1, \dots, \mathcal{A}_m$ ou se $\mathcal{B}_1, \dots, \mathcal{B}_n \not\vdash \mathcal{A}_1, \dots, \mathcal{A}_m$.

O método é baseado em *refutação*, isto é, para provarmos que $\mathcal{B}_1, \dots, \mathcal{B}_n \vdash \mathcal{A}_1, \dots, \mathcal{A}_m$, supomos a falsidade do sequente como um todo, considerando a veracidade de $\mathcal{B}_1, \dots, \mathcal{B}_n$ e a falsidade de $\mathcal{A}_1, \dots, \mathcal{A}_m$, esperando derivarmos uma *contradição*. Se a contradição for obtida, teremos demonstrado o sequente, caso contrário, o método produz uma valoração que satisfaz todas as fórmulas \mathcal{B}_i do antecedente e falsifica todas as fórmulas \mathcal{A}_i do consequente, estabelecendo

um *contra-exemplo* para o sequente.

Para indicar se uma fórmula está sendo considerada verdadeira ou falsa, o método dos tableaux lida com *fórmulas rotuladas* pelos símbolos T (*true*) e F (*false*). O primeiro passo na aplicação do método é falsificar o objetivo inicial e para isto, marcamos as fórmulas do antecedente como T e as do conseqüente como F. As fórmulas são encadeadas, formando o tableaux inicial, conforme representado abaixo:

$$\begin{array}{c}
 \text{T}\mathcal{B}_1 \\
 \vdots \\
 \text{T}\mathcal{B}_n \\
 | \\
 \text{F}\mathcal{A}_1 \\
 \vdots \\
 \text{F}\mathcal{A}_n
 \end{array}$$

O tableaux é uma árvore, que inicialmente possui apenas um ramo. A partir deste ponto, o mesmo é expandido é por regras que podem simplesmente adicionar novas fórmulas ao final de um ramo ou bifurcar um ramo em dois. Escolhemos uma das fórmulas ainda não expandidas do tableaux e escolhemos uma regra apropriada para expandí-la. Marcamos a fórmula que originou a aplicação da regra para indicar que a mesma já foi expandida. A regra aplicada gera novas fórmulas que também podem ser expandidas até que não tenhamos mais fórmulas para expandir no ramo, quando este é considerado um ramo *saturado*. Fórmulas que não podem mais ser expandidas são fórmulas do tipo $\text{T}p$ ou $\text{F}p$, onde p é um dos símbolos proposicionais usado e são consideradas fórmulas *atômicas*. As regras do método de tableaux para a lógica proposicional clássica são as apresentadas na tabela 2.2:

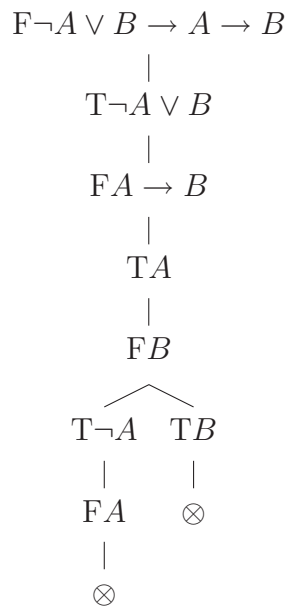
$\text{T}A \rightarrow B$	$\text{T}A \vee B$	$\text{F}A \wedge B$
$\text{F}A \quad \text{T}B$	$\text{T}A \quad \text{T}B$	$\text{F}A \quad \text{F}B$
$\text{F}A \rightarrow B$	$\text{F}A \vee B$	$\text{T}A \wedge B$
 $\text{T}A$	 $\text{F}A$	 $\text{T}A$
 $\text{F}B$	 $\text{F}B$	 $\text{T}B$
$\text{T}\neg A$		$\text{F}\neg A$
 $\text{F}A$		 $\text{T}A$

Tabela 2.2: Regras do Tableaux para Lógica Proposicional Clássica

Como as regras de expansão sempre geram fórmulas de tamanho menor, todas as fórmulas serão expandidas até chegarmos a fórmulas atômicas, quando

todos os ramos estarão saturados. Portanto, o processo de expansão sempre termina. Um ramo é considerado *fechado* se possui um par de fórmulas do tipo TA e FA e, assim, não precisa mais ser expandido, mesmo que ainda não esteja saturado, isto é, mesmo que ainda tenha fórmulas em aberto. Consideramos o tableaux como um todo fechado, se todos os seus ramos estão fechados. Neste caso o objetivo inicial é um teorema.

A prova do nosso exemplo anterior, a fórmula (2-2), usando tableaux, segue abaixo, onde o símbolo de sequente foi interpretado como o conectivo lógico \rightarrow e o símbolo \otimes representa os ramos fechados.



O método de tableaux analíticos, de forma semelhante ao cálculo de sequentes, também é bastante mecanizável e por isto, serve de base para muitos provadores automatizados de teoremas. Na verdade, da mesma forma que relacionamos cálculo de sequentes com dedução natural, podemos facilmente relacionar as regras de tableaux com cálculo de sequentes e mostrar que os dois sistemas também provam os mesmos teoremas. Para isto, consideramos o método de tableaux como um cálculo de sequentes de um lado apenas, isto é, podemos escrever as regras de tableaux em cálculo de sequentes, com fórmulas em que o antecedente é vazio. Abaixo mostramos este mapeamento para a lógica proposicional clássica.

$$\frac{\Rightarrow \Delta, TA, TB}{\Rightarrow \Delta, T(A \wedge B)} \quad \frac{\Rightarrow \Delta, FA \quad \Rightarrow \Delta, FB}{\Rightarrow \Delta, F(A \wedge B)}$$

$$\frac{\Rightarrow \Delta, FA, FB}{\Rightarrow \Delta, F(A \vee B)} \quad \frac{\Rightarrow \Delta, TA \quad \Rightarrow \Delta, TB}{\Rightarrow \Delta, T(A \vee B)}$$

$$\frac{\Rightarrow \Delta, TA, FB}{\Rightarrow \Delta, F(A \rightarrow B)} \quad \frac{\Rightarrow \Delta, FA \quad \Rightarrow \Delta, TB}{\Rightarrow \Delta, T(A \rightarrow B)}$$

$$\frac{\Rightarrow \Delta, TA}{\Rightarrow \Delta, F(\neg A)} \quad \frac{\Rightarrow \Delta, FA}{\Rightarrow \Delta, T(A \neg B)}$$

$$\overline{\Rightarrow \Delta, TA, TB}$$