UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA – UESB
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNÓLOGICAS – DCET
CURSO DE LICENCIATURA EM MATEMÁTICA

JÉFFERSON DOS SANTOS RIBEIRO

ALGUNS MÉTODOS DE PROVAS EM LÓGICA

VITÓRIA DA CONQUISTA – BA
MARÇO DE 2016

JÉFFERSON DOS SANTOS RIBEIRO

ALGUNS MÉTODOS DE PROVAS EM LÓGICA

Trabalho de Conclusão de Curso apresentado à Banca Examinadora da Universidade Estadual do Sudoeste da Bahia – UESB / Campus de Vitória da Conquista – BA, como requisito parcial para obtenção do título de Licenciado em Matemática, sob orientação do Prof. Dr.:Sergio Silva Aguiar.

VITÓRIA DA CONQUISTA

2016

FOLHA DA BIBLIOTECA

FOLHA DE APROVAÇÃO

JÉFFERSON DOS SANTOS RIBEIRO

ALGUNS MÉTODOS DE PROVAS EM LÓGICA

Trabalho de Conclusão de Curso apresentado à banca examinadora da Universidade Estadual do Sudoeste da Bahia – UESB / Campus de Vitória da Conquista – BA, como requisito parcial para obtenção do título de Licenciado em Matemática, sob orientação do Prof. Dr.: Sergio da Silva Aguiar.

Vitória da Conquista, 31 de Março de 2016.

BANCA EXAMINADORA:

_____

Antônio Augusto Oliveira Lima
Universidade Estadual do Sudoeste da Bahia

_____

Eridan da C. S. Maia
Universidade Estadual do Sudoeste da Bahia

_____

Sergio da Silva Aguiar – Orientador
Universidade Estadual do Sudoeste da Bahia

AGRADECIMENTOS

Agradeço primeiramente a professora Eridan, por ser a professora que me fez despertar o interesse em aprofundar o estudo da Lógica durante a disciplina Fundamentos de Matemática Elementar I.

Agradeço aos discentes que participaram das aulas de monitoria nos anos em que fui monitor de FMEI, foi de grande importância para o meu aprendizado.

Agradeço ao professor Sergio Aguiar, pelos ensinamentos durante as aulas quando fui seu aluno, principalmente pelas orientações e pela paciência durante a escrita desde trabalho.

Agradeço aos professores Augusto, Débora, Ana Paula, Jonson, Reginaldo (em memória) e Adilson por todo o apoio durante o curso e pela amizade que construímos.

Agradeço a meus amigos, familiares e colegas de curso que me incentivaram nesse trabalho.

Agradeço as pessoas mais importantes da minha vida, meu pai Miguel, minha mãe Damiana, minha irmã Mariana e a minha sobrinha Paloma, sem o amor deles eu nada seria.

Agradeço a Deus, principalmente, pela liberdade de pensar, de desejar e de optar. De não ser nada, mas poder ter me tornado quem sou.

Enfim, agradeço a todos que direta ou indiretamente contribuíram para que eu chegasse até aqui.

“We're beautiful, like diamonds in the sky”
Rihanna - Diamonds

RESUMO

O presente trabalho faz uma exposição sobre linguagem da Lógica Proposicional e de Predicados com alguns métodos de provas utilizados em computação. Visando complementar o conteúdo de Lógica apresentado no I semestre do curso de Licenciatura em Matemática na Universidade Estadual do Sudoeste da Bahia – Campus de Vitória da Conquista, apresentamos novos conteúdos que envolvem processos de deduções e provas em Lógica. Estes processos dedutivos tem como base métodos formais que podem ser implementados em uma máquina.

Palavras chave: Matemática, Lógica, Provas.

ABSTRACT

This work makes an exhibition about the languages of Propositional Logic and Predicate Logic with some methods of proofs used in computing. To complement the logic of content presented in the first semester of the Bachelor's Degree in Mathematics at the State University of Southwestern Bahia – Vitória da Conquista's campus, we present new contents involving processes of deductions and proofs, in Logic. These deductive processes are based on formal methods that can be implemented in a machine.

Keywords: Mathematics, Logic, Proofs.

SUMÁRIO

INTRODUÇÃO: 10
Lógica Proposicional: 11
Sintaxe 11
Semântica 13
Propriedades Semânticas 14
Prova 19
Lógica de Predicados: 36
Sintaxe 36
Semântica 37
Prova 39
CONSIDERAÇÕES FINAIS: 45
REFERÊNCIAS: 46

INTRODUÇÃO

Lógica é um conteúdo ou disciplina do curso de Matemática que explora algumas teorias, dentre elas a teoria da prova, que é um ramo da Lógica Matemática que considera provas como objetos matemáticos. Desse modo, acredita-se que Lógica pode ser tratada por meio de operações matemáticas.

Este trabalho estende o conteúdo de Lógica que é apresentado aos discentes no primeiro semestre do curso de Licenciatura em Matemática da Universidade Estadual do Sudoeste da Bahia – Campus de Vitória da Conquista, mostrando alguns métodos de prova em Lógica utilizados em computação. Está dividido em duas partes: A primeira explora a Lógica Proposicional apresentando suas dimensões sintática e semântica, bem como sistemas de dedução, tais como tabelas verdade, tableaux semânticos, resolução e sistema de Gentzen. O estudo detalhado dessa lógica proposicional é importante porque contém quase todos os conceitos importantes necessários para o estudo de lógicas mais complexas.

A segunda parte é uma introdução a Lógica de Predicados e estende a linguagem e os sistemas de dedução da Lógica Proposicional à Lógica de Predicados.

Devemos considerar que o conhecimento das diversas técnicas ou maneiras de provar teoremas seja de grande utilidade para o curso de Matemática. Além disso, julgamos que sejam importantes os conhecimentos que relacionam Matemática com Ciência da Computação.

Esperamos assim, contribuir para despertar o interesse de estudantes para lógica formal que é base para um raciocínio organizado e coerente.

Lógica Proposicional

A Lógica Proposicional (L_P) estuda como raciocinar com proposições, isto é, como deduzir de um conjunto de hipóteses consideradas verdadeiras, determinada conclusão verdadeira. Para isso, são fundamentais as noções de proposição, verdade, dedução e prova. Esta lógica leva em conta somente os valores verdades e a forma das proposições.

A linguagem da Lógica Proposicional é uma linguagem formal em que podemos distinguir as dimensões: sintática e semântica. A dimensão sintática diz respeito à escrita correta de fórmulas e a semântica diz respeito ao significado das fórmulas.

Sintaxe

Toda linguagem é constituída de dois elementos básicos: um alfabeto formado por conjuntos de símbolos usados na linguagem e uma gramática que é o conjunto de regras pelas quais são definidas as palavras corretas ou fórmulas bem formadas.

O alfabeto da L_P, $\sum$, é constituído dos seguintes conjuntos disjuntos:

- Conjunto enumerável, $\sum_v$, de variáveis proposicionais, $\sum_v$ = {P, Q, R, S, P_1, ... verdadeiro, falso}.
- Conjunto, $\sum_c$, de conectivos proposicionais, $\sum_c = \{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$.
- Conjunto, $\sum_p$, de símbolos de pontuação $\sum_p$= {); (; ,}.

Assim, o alfabeto da linguagem proposicional $\sum$, pode ser escrito como:

$\sum = \sum_v \cup \sum_c \cup \sum_p$.

Proposição

É toda sentença declarativa, ou seja, de sentido completo ou fechado e que satisfaz as seguintes condições:

- Passível de ser interpretada e valorada.
- Obedece a lei do terceiro excluído: pode ser valorada como verdadeira ou falsa. (Na Lógica Proposicional).
- Obedece a lei de não contradição. Não pode ser simultaneamente valorada como verdadeira e falsa.

Exemplos:

A soma de 3 com 8 é 11.

A capital do Brasil é Brasília.

2 + 2 = 0.

As proposições podem ser:

Simples ou atômicas: aquelas não ligadas por conectivos lógicos. Exemplo: P, Q, R, P_1.

Compostas: aquelas que são formadas pela combinação de duas ou mais proposições simples.

Por meio de conectivos, então, podemos formar novas proposições. Como não nos interessa estudar somente as proposições simples, podemos aplicar recursivamente regras gramaticais para criarmos proposições ou fórmulas compostas.

Fórmulas

As fórmulas da L_P são geradas por $\{\sum, G\}$, onde G é a gramática constituída pelas seguintes regras:

g_1: $\frac{-}{P}$, $P \in \sum_v$

g_2: $\frac{a}{\neg a}$, $a \in L_P$

g_3: $\frac{a,\beta}{a \wedge \beta}$, $a, \beta \in L_P$

g_4: $\frac{a,\beta}{a \vee \beta}$, $a, \beta \in L_P$

g_5: $\frac{a,\beta}{a \rightarrow \beta}$, $a, \beta \in L_P$

g_6: $\frac{a,\beta}{a \leftrightarrow \beta}$, $a, \beta \in L_P$

Entende-se que os objetos que estão acima do traço horizontal ── dão origem aos que estão abaixo. Pela regra g_1, entendemos que P foi gerado a partir de um conjunto vazio, por isso é chamada de axioma da linguagem formal.

Os parênteses, assim como na aritmética, são empregados para priorizar um cálculo proposicional. Esses símbolos podem ser omitidos quando isto não altera o significado da fórmula proposicional.

Exemplo:

$((\neg(\neg P)) \rightarrow Q) \equiv \neg\neg P \rightarrow Q$

Observação: $\neg(P \wedge Q) \neq \neg P \wedge Q$.

A utilização da ordem de precedência permite a simplificação das fórmulas com eliminação dos símbolos de pontuação. Na L_P, a ordem de precedência dos conectivos proposicionais segundo (SOUZA, 2002), é definida por:

- Maior precedência: $\neg$
- Precedência intermediária: $\rightarrow$, $\leftrightarrow$
- Menor precedência: $\wedge$, $\vee$.

Semântica:

A cada fórmula sintática podemos associar um significado. O mundo semântico é onde se define o significado dos símbolos e fórmulas do mundo sintático.

O computador, por exemplo, é uma máquina estritamente sintática, sendo necessário dar um significado ou semântica aos símbolos por ela manipulados. O ato de programar pode ser considerado como a tradução de um conhecimento semântico para um programa sintático que é manipulado pela máquina.

A semântica na Lógica Proposicional é determinada por uma função, a Interpretação (I). Essa função associa um valor verdade a cada fórmula da L_P. Esse valor é representado por V e F. Assim, em L_P as fórmulas possuem dois significados e é denominada lógica bivalente por possuir apenas dois estados semânticos.

A definição da interpretação dos símbolos do alfabeto da Lógica Proposicional é considerada a seguir por (SOUZA, 2002).

Uma interpretação I, na Lógica Proposicional, é uma função binária, onde:

- O domínio de I é formado pelo conjunto de fórmulas da L_P.
- O contradomínio de I é o conjunto {V, F}.
- O valor de I, tendo como argumentos os símbolos de verdade é dado por I[verdadeiro] = V e I[falso] = F.
- Dado um símbolo proposicional P, então I[P] $\in$ {V, F}.

A interpretação dos símbolos verdade é fixa. Isto é, para qualquer interpretação I, todas as interpretações têm a mesma opinião sobre os significados dos símbolos de verdade V e F.

Dadas uma fórmula E e uma interpretação I, então o significado de E, indicado por I[E], é determinado pelas regras:

- Se E = P, onde P é um símbolo proposicional, então I[E] = I[P] e I[P] $\in$ {V, F}.
- Se E = verdadeiro, então I[E] = I[verdadeiro] = V. Se E = falso, então I[E] = I[falso] = F.
- Seja H uma fórmula. Se E = $\neg$H, então

I[E] = I[$\neg$H] = V se I[H] = F e I[E] = I[$\neg$H] = F se I[H] = V.

- Sejam H e G duas fórmulas. Se E = (H $\wedge$ G), então

I[E] = I[H $\wedge$ G] = V se I[H] = V e I[G] = V e I[E] = I[H $\wedge$ G] = F se I[H] = F e/ou I[G] = F.

- Sejam H e G duas fórmulas. Se E = (H $\vee$ G), então

$I[E] = I[H \vee G] = V$ se $I[H] = V$ e/ou $I[G] = V$ e $I[E] = I[H \vee G] = F$ se $I[H] = F$ e $I[G] = F$.

- Sejam H e G duas fórmulas. Se $E = (H \rightarrow G)$, então

$I[E] = I[H \rightarrow G] = V$ se $I[H] = F$ e/ou $I[G] = V$ e $I[E] = I[H \rightarrow G] = F$ se $I[H] = V$ e $I[G] = F$.

- Sejam H e G duas fórmulas. Se $E = (H \leftrightarrow G)$, então

$I[E] = I[H \leftrightarrow G] = V$ se $I[H] = I[G]$ e $I[E] = I[H \leftrightarrow G] = F$ se $I[H] \neq I[G]$.

Os conectivos isolados não tem nenhum significado na LP. As regras semânticas também são representadas por tabelas, chamadas tabelas verdade. As tabelas verdade associadas aos conectivos proposicionais são definidas na tabela a seguir. Neste caso, H e G são fórmulas na LP.

$I[H]$	$I[G]$	$I[\neg H]$	$I[H \wedge G]$	$I[H \vee G]$	$I[H \rightarrow G]$	$I[H \leftrightarrow G]$
V	V	F	V	V	V	V
V	F	F	F	V	F	F
F	V	V	F	V	V	F
F	F	V	F	F	V	V

Podemos perceber na tabela que dada uma interpretação I, se $I[H] = V$ e $I[G] = F$, então $I[\neg H] = F$, $I[H \wedge G] = F$, $I[H \rightarrow G] = F$ e assim por diante.

Nesse exemplo a seguir, veja a construção de uma tabela verdade associada à fórmula H, dada por:

$H = (\neg P \vee Q) \rightarrow (Q \wedge P)$

$I[P]$	$I[Q]$	$I[\neg P]$	$I[\neg P \vee Q]$	$I[Q \wedge P]$	$I[H]$
V	V	F	V	V	V
V	F	F	F	F	V
F	V	V	V	F	F
F	F	V	V	F	F

Observe que a coluna da fórmula H é obtida a partir das colunas intermediárias $\neg P$, $(\neg P \vee Q)$ e $(Q \wedge P)$.

Propriedades Semânticas

Na semântica podemos destacar um conjunto de propriedades fundamentais no estudo da lógica. E esse conjunto é definido a seguir.

- Seja P uma proposição da Linguagem Proposicional, P diz-se uma tautologia se qualquer que seja a atribuição de valores verdade para as proposições atômicas que constituem P, o valor verdade de P for sempre Verdadeiro. Em outras palavras, se para cada atribuição de valores verdade de P, I[P] = V. Isto é, numa tautologia o importante é a forma e não sua interpretação.

Nesse exemplo a seguir, vejamos a construção de uma tabela verdade associada à fórmula H.

$H = P \wedge Q \rightarrow Q$

I[P]	I[Q]	I[P ∧ Q]	I[P ∧ Q → Q]
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	V

Podemos dizer, considerando a tabela acima que H é uma tautologia.

Também podemos fazer a demonstração de uma tautologia usando a interpretação da fórmula.

Exemplo:

Seja a fórmula $G = P \vee \neg P$. Observe que para toda interpretação I, I[G] = V, pois:

$I[G] = V \Leftrightarrow I[P \vee \neg P] = V$

$\Leftrightarrow I[P] = V$ e/ou $I[\neg P] = V$

$\Leftrightarrow I[P] = V$ e/ou $I[P] = F$.

Como $I[G] \in \{V, F\}$, então I[P] = V ou I[P] = F. Portanto, a afirmação I[P] = V e/ou I[P] = F é verdadeira. Logo I[G] = V.

- Uma proposição, M, da Lógica Proposicional é uma contradição se para toda atribuição de valores verdades para as proposições atômicas que constituem M, o valor verdade de M é sempre Falso. Um exemplo de contradição é $M = P \wedge \neg P$. Observe que, M uma tautologia se, e somente se, ¬M for uma contradição e, inversamente, M é uma contradição se, e somente se, ¬M for uma tautologia.

Suponha que exista uma interpretação I tal que I[M] = V.

$I[M] = V \Leftrightarrow I[P \wedge \neg P] = V$

$\Leftrightarrow I[P] = V$ e $I[\neg P] = V$

$\Leftrightarrow$ I[P] = V e I[P] = F.

Como I[M] $\in$ {V, F}, então I[P] = V ou I[P] = F. Logo, a afirmação I[P] = V e I[P] = F é uma afirmação falsa. Portanto I[M] = F.

Por tabela verdade:

I[P]	I[¬P]	I[M]
V	F	F
V	F	F
F	V	F
F	V	F

As fórmulas bem formadas que não são tautologia nem contradição são chamadas de contingentes, pois o valor verdade delas varia de acordo com a interpretação de seus símbolos proposicionais, ou seja, é falsa em alguma interpretação e verdadeira em outra. Uma proposição B da Lógica Proposicional é dita satisfatível se existir uma atribuição de valores verdades para as proposições atômicas de B, tal que I[B] = V, ou seja, se existir uma interpretação onde a fórmula seja verdadeira. Inversamente, uma proposição B da L_P é dita insatisfatível se não for satisfatível, ou seja, se for uma contradição.

Exemplo: Seja a fórmula B = P $\vee$ Q

Considere I e J duas das quatro interpretações para P e Q em B.

I[P] = V, I[Q] = F

J[P] = F, J[Q] = F

I[P]	I[Q]	I[B]
V	V	V
V	F	V
F	V	V
F	F	F

Logo, I[B] = V (2ª linha da tabela) e J[B] = F (última linha da tabela).

Portanto B é satisfatível, mas não é uma tautologia.

- Implicação. Considere as fórmulas:

E = ((P $\wedge$ Q) $\vee$ Q)

H = P $\wedge$ Q

$G = P \rightarrow Q$

Vejamos a tabela verdade associada a estas fórmulas.

I[P]	I[Q]	I[E]	I[H]	I[G]
V	V	V	V	V
V	F	F	F	F
F	V	V	F	V
F	F	F	F	V

Diz-se que uma proposição P implica logicamente ou, simplesmente, implica uma proposição Q, se Q for verdadeira sempre que P for verdadeira. Indicamos $P \Longrightarrow Q$. Como consequência imediata da definição temos que $P \Longrightarrow Q$ significa que a condicional $P \rightarrow Q$ é tautológica, isto é, $P \rightarrow Q \Leftrightarrow V$.

Por definição, se temos que $P \Longrightarrow Q$, então não ocorre à situação I[P] = V e I[Q] = F que é o único caso em que a condicional é falsa. Logo, $P \rightarrow Q$ é tautologia.

Podemos destacar da tabela acima:

- E implica G, pois se I[E] = V, então I[G] = V.
- H implica G, H implica E.
- E não implica em H, pois a terceira linha contém I[E] = V e I[H] = F.
- G não implica E e G não implica H.

Exemplo: Considere as fórmulas:

$M = P \wedge Q$ e $N = P$.

Seja I[M] = V.

$I[M] = V \Leftrightarrow I[P \wedge Q] = V$

$\Leftrightarrow = I[P] = V$ e $I[Q] = V$

$\Longrightarrow = I[P] = V$.

Portanto, se I[M] = V então I[P] = V, ou seja, I[N] = V. Isto é, a fórmula M implica a fórmula N.

Geralmente os teoremas em Matemática são da forma $P \Longrightarrow Q$, isto é, uma condicional $P \rightarrow Q$ tautológica, onde P é chamada de hipótese e Q é a tese.

- Duas fórmulas são equivalentes logicamente, se e somente se os valores verdade obtidos, forem idênticos para cada combinação possível das variáveis que formam as proposições.

Sejam as fórmulas $H = (\neg P \wedge \neg Q)$ e $G = \neg(P \vee Q)$. H e G são equivalentes. (SOUZA, 2002) Essa equivalência estabelece um importante resultado denominado Lei de De Morgan. Dada uma interpretação I temos:

$I[H] = V \Leftrightarrow I[(\neg P \wedge \neg Q] = V$
$\Leftrightarrow I[\neg P] = V$ e $I[\neg Q] = V$
$\Leftrightarrow I[P] = F$ e $I[Q] = F$
$\Leftrightarrow I[P \vee Q] = F$
$\Leftrightarrow I[\neg(P \vee Q)] = V$
$\Leftrightarrow I[G] = V$

$I[H] = F \Leftrightarrow I[(\neg P \wedge \neg Q] = F$
$\Leftrightarrow I[\neg P] = F$ e/ou $I[\neg Q] = F$
$\Leftrightarrow I[P] = V$ e/ou $I[Q] = V$
$\Leftrightarrow I[P \vee Q] = V$
$\Leftrightarrow I[\neg(P \vee Q)] = F$
$\Leftrightarrow I[G] = F$

Logo, $I[H] = I[G]$, isto é, H e G são equivalentes.

Uma outra forma de demonstrar que H é equivalente a G, é construir uma tabela verdade associada H e G e verificar que as suas colunas coincidem.

$I[P]$	$I[Q]$	$I[\neg P \wedge \neg Q]$	$I[\neg(P \vee Q)]$
V	V	F	F
V	F	F	F
F	V	F	F
F	F	V	V

Dadas às fórmulas M e N,

M equivale a N $\Leftrightarrow (M \leftrightarrow N)$ é uma tautologia.

$\Leftrightarrow$ M implica N e N implica M.

Argumento

Argumento é uma afirmação de que um conjunto de proposições simples ou compostas P_1, P_2, ..., P_n chamadas premissas, tem como consequência uma outra proposição simples ou composta Q, chamada conclusão. Ou seja, afirma que determinada conclusão é obtida a partir de premissas dadas. Um argumento será indicado por: $P_1, P_2, ..., P_n \vdash Q$.

Conforme já sabemos, uma proposição pode ser verdadeira ou falsa. No caso de um argumento diremos que ele é válido ou não válido.

Definição: Um argumento é válido se uma conclusão, Q, for verdadeira sempre que as premissas, P_1, P_2, ..., P_n ..., forem verdadeiras, ou seja, se $P_1 \wedge P_2 \wedge ... \wedge P_n \Rightarrow Q$ isto é, $P_1 \wedge P_2 \wedge ... \wedge P_n \rightarrow Q$ é uma tautologia. (Teorema da Dedução).

Os argumentos segundo (MAIA, 2005) se dividem em dois grupos: dedutivos e indutivos. O argumento será dedutivo quando suas premissas fornecem prova conclusiva da veracidade da conclusão, isto é, o argumento é dedutivo quando a conclusão é completamente derivada das premissas. Os argumentos dedutivos são os que nos interessa.

A maioria das propriedades até aqui foram demonstradas pelo método da tabela verdade, que tem um mecanismo bastante simples para verificar a validade de um argumento. Tabelas verdade nos fornecem algoritmos para responder questões relativas aos conectivos vistos como funções booleanas, tais como se uma dada proposição é uma tautologia, contradição ou contingente, se uma proposição for consequência lógica ou equivalente a uma outra. Entretanto, dependendo do tamanho da fórmula, sua construção e análise pode ser uma tarefa com alto custo computacional e, portanto, inviável. De modo geral, se uma fórmula contiver *n* símbolos proposicionais distintos, a tabela verdade terá 2^n linhas (uma linha para cada interpretação possível). Por exemplo, a tabela verdade para o argumento $\{p \rightarrow q, \neg p \rightarrow r, q \rightarrow s, \neg s\} \vdash r$ terá $2^4 = 16$ linhas. Assim, quando o número de proposições num argumento for muito grande, um método mais eficiente para sua validação é necessário. O que veremos a seguir não será tratado com uso de tabelas verdade. A abordagem será feita por meio de uma teoria formal ou método formal.

No emprego do método dedutivo desempenham papéis importantes as equivalências semânticas entre fórmulas. Fórmulas podem ser substituídas por fórmulas semanticamente equivalentes.

Prova

Uma prova de uma fórmula φ, a partir de um conjunto de fórmulas Δ, segundo (Pereira, 2011), consiste numa sequência finita de fórmulas $\gamma_1,\gamma_2,...,\gamma_n$, onde $\gamma_n = \varphi$ e cada γ_i é uma fórmula em Δ ou é derivada de fórmulas em $\Delta \cup \{\gamma_1,...,\gamma_{i-1}\}$, por meio de uma regra de inferência.

As regras de inferência são implicações semânticas. Elas são utilizadas para fazer inferências, ou seja, executar os passos de uma prova ou dedução.

Usamos a notação $\Delta \vdash \varphi$ para indicar que a fórmula φ pode ser derivada a partir das fórmulas em Δ (ou seja, que é possível provar φ a partir de Δ). Uma regra de inferência é um padrão que estabelece como uma nova fórmula pode ser gerada a partir de outras.

Dado um conjunto de fórmulas Δ, uma regra de inferência é correta se permite derivar apenas fórmulas que são consequências lógicas de Δ e é completa se permite derivar todas as fórmulas que são consequências lógicas de Δ. As regras de inferência clássicas são corretas e completas para todo conjunto consistente de fórmulas bem formadas da lógica proposicional.

O processo de prova por dedução consiste em demonstrar que, dadas algumas expressões como verdadeiras (hipóteses ou premissas), uma nova sentença também é verdadeira. Quando isso ocorre, dizemos que a sentença provada é um teorema com respeito às hipóteses.

A prova de um teorema consiste em derivar a expressão desejada H a partir das hipóteses β, utilizando os recursos disponíveis em algum dos sistemas de dedução válidos ($\beta \vdash H$).

Definição: Uma prova formal é uma sequência que mostra que uma proposição segue de proposições anteriores, utilizando regras de inferência. Em outras palavras, que proposições (conclusões) seguem de premissas "verdadeiras".

Sistemas de dedução, também denominados sistemas formais, são completos (se $\beta \Rightarrow H$, então $\beta \vdash H$) e corretos (se $\beta \vdash H$, então $\beta \Rightarrow H$) e estabelecem estruturas que permitem a representação e dedução do conhecimento.

Os sistemas de dedução podem ser divididos em dois grupos, como segue:

• Sistemas de difícil implementação computacional:

- Sistema Axiomático
- Dedução Natural

• Sistemas mais adequados para implementação computacional:

- Tableaux Semânticos
- Resolução
- Sistema de Gentzen

Apresentaremos a seguir os métodos mais adequados a para implementação computacional.

Tableaux Semânticos

O método de tableau para a lógica clássica, devido a Jaakko Hintikka (1929-2015) e Evert Beth (1908-1964) em [Hin55, Bet59], é uma prova por redução ao absurdo. Começamos com uma hipótese e se a partir dessa hipótese derivamos uma consequência falsa, então podemos concluir que a hipótese original é falsa. Se a prova não foi bem sucedida, e nenhuma consequência falsa é vislumbrada, então em alguns casos podemos concluir que não existe erro com a hipótese aberta, isto é, a hipótese pode ser verdadeira para alguma interpretação. Existem casos nos quais podemos mostrar que a procura por uma consequência falsa foi exaustiva, e poderíamos ter encontrado uma consequência falsa se uma existisse. Mas nem todos os casos são assim. Algumas vezes tudo o que podemos dizer sobre a prova é que ainda não chegamos a nenhuma consequência falsa e que não sabemos se isto irá acontecer caso continuemos a prova.

Um tableau semântico na Lógica Proposicional segundo (SOUZA, 2002) é uma sequência de fórmulas construída de acordo com certas regras e geralmente apresentada sob a forma de uma árvore. Os elementos básicos que fazem parte dessa árvore são definidos pela composição dos elementos:

- O alfabeto da Lógica Proposicional.
- O conjunto das fórmulas da LP.
- Um conjunto de regras de dedução.

O tableau semântico contém apenas regras de dedução, que definem o mecanismo de inferência, permitindo a dedução de conhecimento.

As regras de inferência do tableau semântico são definidas da seguinte maneira: Sejam A e B duas fórmulas da Lógica Proposicional. As regras de inferência do tableau semântico na LP são de R_1 a R_9 indicadas abaixo.

$R_1 = A \wedge B$

A

B

$R_2 = A \vee B$

$A \quad B$

$R_3 = A \rightarrow B$

$\neg A \quad B$

$R_4 = A \leftrightarrow B$

$A \wedge B \quad \neg A \wedge \neg B$

$R_5 = \neg\neg A$

A

$R_6 = \neg(A \wedge B)$

$\neg A \quad \neg B$

$R_7 = \neg(A \vee B)$

$\neg A$

$\neg B$

$R_8 = \neg(A \rightarrow B)$

A

$\neg B$

$R_9 = \neg(A \leftrightarrow B)$

$\neg A \wedge B \quad A \wedge \neg B$

O método de prova nos tableaux semânticos é feito utilizando o método da negação ou absurdo. Assim, para provar a fórmula A, é considerada inicialmente a sua negação $\neg A$. A partir de $\neg A$, o tableau é construído utilizando as regras de dedução, cuja aplicação decompõe a fórmula $\neg A$ em subfórmulas.

Exemplo. Considere o conjunto de fórmulas: $\{(A \vee B), (A \wedge \neg B)\}$.

.1	$A \vee B$		
.2	$A \wedge \neg B$		
.3	A	B	R_2, .1
.4	A	A	R_1, .2
.5	$\neg B$	$\neg B$	R_1, .2

Na construção de um tableau semântico aplique preferencialmente as regras R_1, R_5, R_7 e R_8, que não bifurcam o tableau.

Considere o exemplo anterior.

.1	$A \vee B$		
.2	$A \wedge \neg B$		
.3	A		R_1, .2
.4	$\neg B$		R_1, .2
.5	A	B	R_2, .1

Um tableau semântico, na Lógica Proposicional, é construído como se segue. Seja $\{A_1, A_2, ..., A_n\}$ um conjunto de fórmulas.

- A árvore, com apenas um ramo, é um tableau associado a $\{A_1, A_2, ..., A_n\}$.

.1	A_1
.2	A_2
	$\vdots$
.n	A_n

- Seja Tree um tableau associado a $\{A_1, A_2, ..., A_n\}$. Se Tree* é a árvore resultante da aplicação de uma das regras R_1, ..., R_9 à árvore Tree, então Tree* é também um tableau associado a $\{A_1, A_2, ..., A_n\}$.

Exemplo: Considere o conjunto de fórmulas.

$\{(A \rightarrow B), \neg(A \vee B), \neg(C \rightarrow A)\}$.

A árvore de apenas um ramo $Tree^1$, é o tableau associado ao conjunto de fórmulas:

.1 $(A \rightarrow B)$

.2 $\neg(A \vee B)$

.3 $\neg(C \rightarrow A)$

Aplicando R_7 em .2 a $Tree^1$ obteremos $Tree^2$.

.1 $(A \rightarrow B)$

.2 $\neg(A \vee B)$

.3 $\neg(C \rightarrow A)$

.4 $\neg A$ R_7, .2

.5 $\neg B$ R_7, .2

Aplicando uma outra regra a $Tree^2$, obtermos $Tree^3$, e assim por diante.

Um ramo em um tableau é fechado se ele contém uma fórmula A e sua negação ¬A. Um ramo é aberto quando não é fechado. E um tableau é fechado quando todos os seus ramos são fechados. Caso contrário, ele é aberto.

Os tableaux semânticos definem uma estrutura para representação e dedução de conhecimento. Utilizando as regras de inferência, são construídas árvores que determinam um mecanismo de inferência. É definido a seguir a prova de uma fórmula utilizando tableau semântico.

Seja H uma fórmula. Uma prova de H utilizando tableaux semânticos é um tableau fechado associado a ¬H. Neste caso, H é um teorema do sistema de tableaux semânticos.

Exemplo: Considere a fórmula $H = \neg((P \rightarrow Q) \wedge \neg(P \leftrightarrow Q) \wedge P)$.

Construção do tableau associado a H

.1 $\neg\neg((P \rightarrow Q) \wedge \neg(P \leftrightarrow Q) \wedge P)$ $\neg H$

.2 $(P \rightarrow Q) \wedge \neg(P \leftrightarrow Q) \wedge P)$ R_5, .1

.3 $P \rightarrow Q$ R_1, .2

.4 $\neg(P \leftrightarrow Q)$ R_1, .2

.5 P R_1, .2

.6 $\neg P_{fech.}$ Q R_3, .3

.7 $\neg P \wedge Q$ $P \wedge \neg Q$ R_9, .4

.8 $\neg P$ P R_1, .7

.9 $Q_{fech.}$ $\neg Q_{fech.}$ R_1, .7

Observe que o tableau é inicializado com a negação da fórmula H, que é igual a $\neg\neg((P \rightarrow Q) \wedge \neg(P \leftrightarrow Q) \wedge P)$. Após a aplicação das regras, todos os seus ramos são fechados, o que constitui uma prova de H.

Considere agora a fórmula $G = (\neg(P \leftrightarrow Q) \vee P)$.

A construção do tableau associado à fórmula G é dado por:

.1	$\neg(\neg(P \leftrightarrow Q) \vee P)$		$\neg G$
.2	$P \leftrightarrow Q$		R_7, .1
.3	$\neg P$		R_7, .1
.4	$P \wedge Q$	$\neg P \wedge \neg Q$	R_4, .2
.5	$P_{fech.}$	$\neg P$	R_1, .4
.6	Q	$\neg Q_{aberto}$	R_1, .4

Portanto, não se tem uma prova de G, utilizando tableaux semânticos.

Seja H uma fórmula da Lógica Proposicional. Se H for uma tautologia então existe uma prova de H no sistema de tableaux semânticos.

O sistema de tableaux semânticos também é correto. Todo teorema é uma tautologia. Isto significa que dada uma fórmula H, se o tableau associado a $\neg H$ é fechado, então H é uma tautologia. Ou seja, os argumentos provados utilizando tableaux semânticos são válidos.

Seja H uma fórmula da Lógica Proposicional. Se existir uma prova de H utilizando tableaux semânticos, então H é uma tautologia.

Dada uma fórmula H e um conjunto de hipóteses $\beta = \{A_1, A_2, ..., A_n\}$, então H é uma consequência lógica de β, nos tableaux semânticos, se existe uma prova de $(A_1 \wedge A_2 \wedge ... \wedge A_n) \rightarrow H$ utilizando tableaux semânticos.

Notação: $\beta \vdash H$ ou $\{A_1, A_2, ..., A_n\} \vdash H$

Exemplo retirado de (SOUZA, 2002, p. 140) Considere os argumentos.

Guga é determinado.

Guga é inteligente.

Se Guga é determinado e atleta, ele não é um perdedor.

Guga é atleta se é um amante do tênis.

Guga é amante do tênis se é inteligente.

A afirmação “Guga não é um perdedor” é uma consequência lógica dos argumentos acima?

Uma prova utilizando tableaux semânticos é feita para responder essa questão. Considere inicialmente as seguintes correspondências:

$P \equiv$ Guga é determinado,

$Q \equiv$ Guga é inteligente,

$R \equiv$ Guga é atleta,

$P_1 \equiv$ Guga é um perdedor

$Q_1 \equiv$ Guga é amante do tênis.

A partir de tais correspondências, os argumentos são traduzidos pela Lógica Proposicional utilizando o teorema da dedução.

$H = (P \wedge Q \wedge ((P \wedge R) \rightarrow \neg P_1) \wedge (Q_1 \rightarrow R) \wedge (Q \rightarrow Q_1)) \rightarrow \neg P_1$.

Deve-se então provar se H é ou não uma tautologia, ou seja, provar se $\vdash H$. Vejamos.

.1	$\neg((P \wedge Q \wedge ((P \wedge R) \rightarrow \neg P_1) \wedge (Q_1 \rightarrow R) \wedge (Q \rightarrow Q_1)) \rightarrow \neg P_1)$	$\neg H$
.2	$(P \wedge Q \wedge ((P \wedge R) \rightarrow \neg P_1) \wedge (Q_1 \rightarrow R) \wedge (Q \rightarrow Q_1)$	R_8, .1
.3	$\neg\neg P_1$	R_8, .1
.4	P	R_1, .2
.5	Q	R_1, .2
.6	$(P \wedge R) \rightarrow \neg P_1$	R_1, .2
.7	$(Q_1 \rightarrow R)$	R_1, .2
.8	$(Q \rightarrow Q_1)$	R_1, .2
.9	P_1	R_5, .3
.10	$\neg Q_{fech.}$ Q_1	R_3, .8
.11	$\neg Q_{1fech.}$ R	R_3, .7
.12	$\neg(P \wedge R)$ $\neg P_{1fech.}$	R_3, .6
.13	$\neg P_{fech.}$ $\neg R_{fech.}$	R_6, .12

Como o tableau é fechado, então H é uma tautologia e a consequência lógica ocorre.

Exemplo: Seja $\beta = \{\neg A \vee B, \neg(B \vee \neg C), C \rightarrow D, \neg(\neg A \vee D)\}$.

Neste exemplo vamos provar utilizando tableaux semânticos, que o conjunto de fórmulas β é insatisfatível.

Note que β é insatisfatível se e somente se não existe interpretação I tal que:

$I[\neg A \vee B] = I[\neg(B \vee \neg C)] = I[C \rightarrow D] = I[\neg(\neg A \vee D)] = V \Leftrightarrow$

$\forall I, I[(\neg A \vee B) \wedge (B \vee \neg C) \wedge (C \rightarrow D) \wedge \neg(\neg A \vee D)] = F \Leftrightarrow$

$\forall I, I[\neg((\neg A \vee B) \wedge (B \vee \neg C) \wedge (C \rightarrow D) \wedge \neg(\neg A \vee D))] = V \Leftrightarrow$

$H = \neg((\neg A \vee B) \wedge (B \vee \neg C) \wedge (C \rightarrow D) \wedge \neg(\neg A \vee D))$ é uma tautologia.

O tableau referente à fórmula H é dado por:

.1	$\neg\neg((\neg A \vee B) \wedge \neg(B \vee \neg C) \wedge (C \rightarrow D) \wedge \neg(\neg A \vee D))$	$\neg H$
.2	$(\neg A \vee B) \wedge \neg(B \vee \neg C) \wedge (C \rightarrow D) \wedge \neg(\neg A \vee D)$	R_5, .1
.3	$\neg A \vee B$	R_1, .2
.4	$\neg(B \vee \neg C)$	R_1, .2
.5	$C \rightarrow D$	R_1, .2
.6	$\neg(\neg A \vee D)$	R_1, .2
.7	$\neg B$	R_7, .4
.8	$\neg\neg C$	R_7, .4
.9	$\neg\neg A$	R_7, .6
.10	$\neg D$	R_7, .6
.11	C	R_5, .8
.12	A	R_5, .9
.13	$\neg C_{fech.}$ $D_{fech.}$	R_3, .5

Como o tableau é fechado, então $\vdash H$, logo H é uma tautologia. Portanto, o conjunto de fórmulas β é insatisfatível.

Resolução

A resolução na Lógica Proposicional é um método de prova desenvolvido por Robinson nos anos 60 e desde então houve inúmeras pesquisas e implementações utilizando tal método. A resolução segundo (SOUZA, 2002) pode ser considerada como o dual dos tableaux semânticos. Note que nos tableaux semânticos são aplicadas preferencialmente as regras R_1, R_5, R_7 e R_8, que não bifurcam o tableau. Isto significa que o tableau é construído de forma eficiente para fórmulas que são conjunções de disjunções literais. Neste caso, a regra R_1 é aplicada inicialmente e em seguida a R_2. De forma dual, a resolução se aplica a fórmulas que são conjunções de disjunções de literais, representadas na forma e conjunto e cláusulas.

Considere a fórmula H, que é uma conjunção de disjunção de literais.

$H = (P \vee \neg Q \vee R) \wedge (P \vee \neg Q) \wedge (P \vee P)$

Esta fórmula é representada na forma de conjuntos. Da forma

$H = \{\{P, \neg Q, R\}, \{P, \neg Q\}, \{P\}\}$

Onde H é um conjunto de conjunto de literais. As vírgulas mais internas representam o conectivo $\vee$ e as mais externas o conectivo $\wedge$.

Uma cláusula, na Linguagem Proposicional, é uma disjunção de literais. Utilizando a notação de conjuntos, uma cláusula é um conjunto finito de literais. De H podemos destacar as seguintes cláusulas: $C_1 = \{P, \neg Q, R\}$, $C_2 = \{P, \neg Q\}$, $C_3 = \{P\}$.

Dois literais são complementares se um é a negação do outro.

Considere duas cláusulas $C_1 = \{P, \neg Q, R\}$ e $C_2 = \{\neg P, R\}$.

O resolvente de C_1 e C_2 é dado por $res(C_1, C_2) = \{\neg Q, R\}$. Perceba que o resolvente destas duas cláusulas é também uma cláusula.

Agora as seguintes cláusulas $D_1 = \{P, \neg Q\}$ e $D_2 = \{\neg P, Q\}$.

$res(D_1, D_2) = \{\}$, temos neste caso, que a regra de resolução elimina todos os literais de cláusulas pois eles são complementares, a resolução de D_1 e D_2 também é uma cláusula, porém vazia.

O método da resolução na Lógica Proposicional, como tableau semântico, utiliza a linguagem da Lógica Proposicional como elemento básico. A partir desta linguagem, a resolução considera o conjunto das cláusulas da Lógica Proposicional a regra de resolução.

Os elementos básicos da resolução, na L_P, são definidos pela composição dos elementos:

- O alfabeto da Lógica Proposicional.
- O conjunto de cláusulas da Lógica Proposicional.
- A regra de resolução

A resolução é um método de dedução, que também define uma estrutura para representação e dedução do conhecimento. Como nos tableaux semânticos, não tem axiomas, mas apenas uma regra de inferência. A regra de resolução é utilizada na prova de argumentos a partir do conhecimento representado no sistema. A prova é feita aplicando a regra de resolução sobre um conjunto de cláusulas.

O mecanismo de inferência da resolução utiliza apenas uma regra.

Dadas duas cláusulas: $C_1 = \{A_1, A_2, ..., A_n\}$, $C_2 = \{B_1, B_2, ..., B_n\}$.

A regra de resolução aplicada a C_1 e C_2 é definida pelo processo:

Tendo, C_1 e C_2 deduza res(C_1, C_2).

Para provar a satisfatibilidade de um conjunto de cláusulas, a regra de resolução é aplicada repetidamente, até que se consiga a cláusula vazia. Esta aplicação da regra de resolução determina uma expansão por resolução.

Exemplo: Considere o conjunto de cláusulas:

{{¬P, Q, R}, {P, R}, {P, ¬R}}

Uma expansão por resolução sobre este conjunto é feita da seguinte forma:

.1	{¬P, Q, R}	
.2	{P, R}	
.3	{P, ¬R}	
.4	{Q, R}	res(.1, .2)
.5	{Q, P}	res(.3, .4)
.6	{P}	res(.2, .3)

A expansão foi obtida por três aplicações da regra de resolução. Observe que neste caso não é possível obter uma cláusula vazia.

A construção de uma expansão por resolução é dada por:

Seja $\{A_1, A_2, ..., A_n\}$ um conjunto de cláusulas

.1	A_1
.2	A_2
	⋮
.n	A_n

Essa estrutura é uma expansão por resolução sobre $\{A_1, A_2, ..., A_n\}$.

Seja Exp uma expansão por resolução sobre $\{A_1, A_2, ..., A_n\}$. Se Exp* é a estrutura resultante da adição de Res(A_i, A_j), i, j $\leq$ n, i $\neq$ j, à expansão Exp, então Exp* é também uma expansão por resolução sobre $\{A_1, A_2, ..., A_n\}$.

Exemplo: Considere o conjunto de cláusulas:

{{¬P, Q}, {P, R}, {P, ¬Q}, {¬Q, ¬R}}.

Vejamos uma expansão por resolução sobre este conjunto.

.1	{¬P, Q}	
.2	{P, R}	
.3	{P, ¬Q}	
.4	{¬Q, ¬R}	
.5	{Q, R}	res(.1, .2)
.6	{ }	res(.4, .5)

A expansão por resolução resultante contém uma cláusula vazia. A obtenção da cláusula vazia é uma propriedade importante, é análogo à obtenção de um tableau semântico fechado.

Uma expansão por resolução é fechada se ela contém uma cláusula vazia.

A consequência lógica na resolução é definida de forma análoga à consequência nos tableaux semânticos. Têm-se conceitos análogos onde tableaux semânticos são substituídos por expansões por resolução.

Dada um fórmula H, a forma clausal associada a H é uma fórmula Hc tal que Hc é uma conjunção de cláusulas e Hc é equivalente a H.

Toda fórmula proposicional possui uma forma clausal associada. Assim, dado H é possível determinar um conjunção de cláusulas equivalente Hc.

Seja H uma fórmula e ¬Hc a forma clausal associada a ¬H. Uma prova de H por resolução é uma expansão por resolução fechada sobre ¬Hc. Neste caso, H é um teorema do sistema de resolução.

Exemplo: Considere as fórmulas

$H = ((P_1 \vee P_2 \vee P_3) \wedge (P_1 \rightarrow P_4) \wedge (P_2 \rightarrow P_4) \wedge (P_3 \rightarrow P_4)) \rightarrow P_4$

Vamos provar a fórmula H, utilizando a resolução.

O primeiro passo é determinar uma forma clausal ¬Hc associada a ¬H. A fórmula ¬Hc é determinada considerando as seguintes equivalências:

$\neg H = \neg((P_1 \vee P_2 \vee P_3) \wedge (P_1 \rightarrow P_4) \wedge (P_2 \rightarrow P_4) \wedge (P_3 \rightarrow P_4)) \rightarrow P_4)$ equivale a

$\neg(\neg((P_1 \vee P_2 \vee P_3) \wedge (\neg P_1 \vee P_4) \wedge (\neg P_2 \vee P_4) \wedge (\neg P_3 \vee P_4)) \vee P_4)$ equivale a

$(P_1 \vee P_2 \vee P_3) \wedge (\neg P_1 \vee P_4) \wedge (\neg P_2 \vee P_4) \wedge (\neg P_3 \vee P_4) \wedge P_4 = \neg Hc$.

Note que ¬Hc é um conjunção de cláusulas, que pode ser representa como notação de conjuntos. Logo

$\neg Hc = \{\{P_1, P_2, P_3\}, \{\neg P_1, P_4\}, \{\neg P_2, P_4\}, \{\neg P_3, P_4\}, \{\neg P_4\}\}$

Desenvolvendo uma expansão por resolução sobre ¬Hc, temos

.1	$\{P_1, P_2, P_3\}$	
.2	$\{\neg P_1, P_4\}$	
.3	$\{\neg P_2, P_4\}$	
.4	$\{\neg P_3, P_4\}$	
.5	$\{\neg P_4\}$	
.6	$\{P_1, P_3, P_4\}$	res(.1, .3)
.7	$\{P_3, P_4\}$	res(.2, .6)

.8 $\{P_4\}$ res(.4, .7)

.9 $\{\ \}$ res(.5, .8)

A expansão por resolução obtida é fechada, o que constitui uma prova de H.

Exemplo: Considere as fórmulas

$G = ((P_1 \vee P_2) \wedge (P_1 \rightarrow P_4) \wedge (P_2 \rightarrow P_4) \wedge (P_3 \rightarrow P_4)) \rightarrow P_3$

Determinando ¬Gc associada a ¬G

$\neg G = \neg(((P_1 \vee P_2) \wedge (P_1 \rightarrow P_4) \wedge (P_2 \rightarrow P_4) \wedge (P_3 \rightarrow P_4)) \rightarrow P_3)$ equivale a

$\neg(\neg((P_1 \vee P_2) \wedge (\neg P_1 \vee P_4) \wedge (\neg P_2 \vee P_4) \wedge (\neg P_3 \vee P_4)) \vee P_3)$ equivale a

$(P_1 \vee P_2) \wedge (\neg P_1 \vee P_4) \wedge (\neg P_2 \vee P_4) \wedge (\neg P_3 \vee P_4) \wedge \neg P_3 = \neg Gc$.

Representando ¬Gc em notação de conjuntos temos que

$\neg Gc = \{\{P_1, P_2\}, \{\neg P_1, P_4\}, \{\neg P_2, P_4\}, \{\neg P_3, P_4\}, \{\neg P_3\}\}$

Fazendo uma expansão por resolução sobre ¬Gc, temos

.1 $\{P_1, P_2\}$

.2 $\{\neg P_1, P_4\}$

.3 $\{\neg P_2, P_4\}$

.4 $\{\neg P_3, P_4\}$

.5 $\{\neg P_3\}$

.6 $\{P_1, P_4\}$ res(.1, .3)

.7 $\{P_4\}$ res(.2, .6)

A expansão por resolução obtida não é fechada. Portanto, não se tem a prova de G.

Seja H uma fórmula da Lógica Proposicional. Se H é uma tautologia, então existe uma prova de H por resolução. Isto significa que dada uma fórmula H, se uma expansão por resolução associada a ¬H é fechada, então H é uma tautologia. Os argumentos provados utilizando a resolução são válidos.

Seja H uma fórmula da Lógica Proposicional. Se existe uma prova de H, por resolução, então H é uma tautologia.

Dada uma fórmula H e um conjunto de hipóteses $\beta = \{A_1, A_2, ..., A_n\}$, então H é uma consequência logica de β, por resolução, se existe uma prova de $(A_1 \wedge A_2 \wedge ... \wedge A_n) \rightarrow H$ por resolução.

Notação $\beta \vdash H$ ou $\{A_1, A_2, ..., A_n\} \vdash H$

Exemplo retirado de (SOUZA, 2002, p. 150). Considere os argumentos.

Guga é determinado.

Guga é inteligente.

Se Guga é determinado e atleta, ele não é um perdedor.

Guga é atleta se é um amante do tênis.

Guga é amante do tênis se é inteligente.

A afirmação "Guga não é um perdedor" é uma consequência lógica dos argumentos acima?

Utilizaremos a resolução para provar que a consequência lógica ocorre.

Representando os argumentos na Lógica Proposicional, temos:

$H = (P \wedge Q \wedge ((P \wedge R) \rightarrow \neg P_1) \wedge (Q_1 \rightarrow R) \wedge (Q \rightarrow Q_1)) \rightarrow \neg P_1$. Onde,

$P \equiv$ Guga é determinado,

$Q \equiv$ Guga é inteligente,

$R \equiv$ Guga é atleta,

$P_1 \equiv$ Guga é um perdedor e

$Q_1 \equiv$ Guga é amante do tênis.

A forma clausal associada a $\neg H$ é obtida pelas seguintes equivalências.

$\neg H = \neg((P \wedge Q \wedge ((P \wedge R) \rightarrow \neg P_1) \wedge (Q_1 \rightarrow R) \wedge (Q \rightarrow Q_1)) \rightarrow \neg P_1)$ equivale a

$\neg(\neg(P \wedge Q \wedge ((P \wedge R) \rightarrow \neg P_1) \wedge (Q_1 \rightarrow R) \wedge (Q \rightarrow Q_1)) \vee \neg P_1)$ equivale a

$P \wedge Q \wedge ((P \wedge R) \rightarrow \neg P_1) \wedge (Q_1 \rightarrow R) \wedge (Q \rightarrow Q_1)) \wedge P_1$ equivale a

$P \wedge Q \wedge (\neg(P \wedge R) \vee \neg P_1) \wedge (\neg Q_1 \vee R) \wedge (\neg Q \vee Q_1)) \wedge P_1$ equivale a

$P \wedge Q \wedge (\neg P \vee \neg R \vee \neg P_1) \wedge (\neg Q_1 \vee R) \wedge (\neg Q \vee Q_1) \wedge P_1 = \neg Hc$.

Colocando $\neg Hc$ em notação de conjunto temos,

$\neg Hc = \{P, Q, \{\neg P, \neg R, \neg P_1\}, \{\neg Q_1, R\}, \{\neg Q, Q_1\}, P_1\}$

Uma expansão por resolução sobre $\neg Hc$ é dada por

.1	$\{P\}$	
.2	$\{Q\}$	
.3	$\{\neg P, \neg R, \neg P_1\}$	
.4	$\{\neg Q_1, R\}$	
.5	$\{\neg Q, Q_1\}$	
.6	$\{P_1\}$	
.7	$\{\neg P, \neg R\}$	res(.3, .6)
.8	$\{\neg Q, R\}$	res(.4, .5)
.9	$\{\neg R\}$	res(.1, .7)
.10	$\{\neg Q\}$	res(.8, .9)
.11	$\{\ \}$	res(.2, .10)

A expansão obtida é fechada, portando $\vdash$ H. Logo, H é uma tautologia.

Exemplo: Este exemplo demonstra, utilizando a resolução que o conjunto de fórmulas β é insatisfatível.

Seja $\beta = \{\neg P \vee Q, \neg(Q \vee \neg R), R \rightarrow P_1, \neg(\neg P \vee P_1)\}$

β é insatisfatível se e somente se $G = \neg((\neg P \vee Q) \wedge \neg(Q \vee \neg R) \wedge (R \rightarrow P_1) \wedge \neg(\neg P \vee P_1))$ é tautologia. Mas G é uma tautologia se e somente se a negação de sua forma clausal ¬Gc tem uma expansão por resolução fechada. Assim ¬Gc é dada por:

$\neg Gc = (\neg P \vee Q) \wedge \neg Q \wedge \neg R \wedge (\neg R \vee P_1) \wedge P \wedge \neg P_1$

$\neg Gc = \{\{\neg P, Q\}, \{\neg Q\}, \{R\}, \{\neg R, P_1\}, \{P\}, \{\neg P_1\}\}$

E sua expansão por resolução é:

.1	$\{\neg P, Q\}$	
.2	$\{\neg Q\}$	
.3	$\{R\}$	
.4	$\{\neg R, P_1\}$	
.5	$\{P\}$	
.6	$\{\neg P_1\}$	
.7	$\{\neg P\}$	res(.1, .2)
.8	$\{\ \}$	res(.5, .7)

A expansão obtida é fechada. Logo $\vdash$ G e pelo teorema da correção, G é uma tautologia. Portanto, β é insatisfatível.

As provas por resolução são análogas às provas que utilizam tableaux semânticos. Nos dois casos é utilizado o método da negação ou absurdo. Na resolução, a obtenção da cláusula vazia corresponde à expansão fechada que é o dual do tableau semântico fechado. Caso não fosse possível obter uma expansão por resolução sobre ¬G, G não seria tautologia.

Sistema de Gentzen

Um sequente é um par ordenado de listas finitas de fórmulas, que pode ser interpretado como que uma das fórmulas da segunda pode ser deduzida da primeira lista. Na prática usamos a notação sugerida por Gentzen: O sequente $\langle \Gamma, \Delta \rangle$ será denotado por $\Gamma \vdash \Delta$.

Uma prova segundo (BEDREGAL; ACIÓLY, 2007) é uma sequência de fórmulas, as quais estabelecem algum sequente com o lado esquerdo sendo composto de todas as hipóteses que não foram descartadas na prova e com o lado direito sendo composto pela fórmula que foi provada ao final da prova. Também, consideraremos as regras de inferências como regras sobre sequentes. Uma prova sempre iniciará com uma assunção, por exemplo, α, e se não acrescentamos nenhuma outra, então ela mesma é considerada como a prova do sequente $\alpha \vdash \alpha$. Assim, a regra que nos permite começar é uma regra a qual nos leva diretamente a que todos os sequentes deste tipo sejam corretos. As outras regras são todas condicionais, elas refletem a ideia de que se certos sequentes são corretos, então os sequentes que podemos concluir deles também são corretos. Por exemplo, a regra Modus Ponens pode ser escrita em forma de sequentes como segue

$\Gamma \vdash \alpha$ e $\Delta \vdash \alpha \rightarrow \beta$ então $\Gamma, \Delta \vdash \beta$.

O que acontece numa prova é que começamos com certos sequentes conhecidos como corretos e então deduzimos que outros sequentes devem, portanto, também ser corretos. A ideia de um cálculo de sequente é manter um registro explícito de que sequente é estabelecido em cada ponto da prova. Isto é possível introduzindo um novo tipo de prova na qual cada linha, por si mesmo, é um sequente provado até esse ponto da prova.

Podemos pensar nas listas que compõem um sequente como tacitamente fechadas por parênteses de chave, pelo que seu papel seria especificar um conjunto de fórmulas. Para isso precisamos considerar duas novas regras de inferência que reflitam estas propriedades intrínsecas a conjuntos, a regra de intercâmbio (Int) e a regra de contração (Con).

Podemos apresentar o cálculo de sequentes como um sistema cujas provas têm uma estrutura de uma árvore, de acordo com as regras. No estilo de Gentzen teremos dois tipos de regras: aquelas que dependem de um dos conectivos lógicos (usadas para introduzir este conectivo ou no lado esquerdo ou direito do sequente) e que por isso são chamadas de regras lógicas e aquelas que não dependem dos conectivos lógicos, mas da disposição ou estrutura do sequente, denominadas regras estruturais. Temos cinco regras estruturais, algumas das quais tem sua versão para o lado direito (que denotaremos por d) e outra para o lado esquerdo (denotado por e), as quais se denominam: inclusão (Inc), enfraquecimento (Enf), corte (Cor), intercâmbio (Int) e contração (Con).

Regras Estruturais:

Inc: $\dfrac{}{\Gamma,\alpha \vdash \alpha,\Delta}$

Cor: $\dfrac{\Gamma \vdash \alpha \quad \Gamma,\alpha \vdash \Delta}{\Gamma \vdash \Delta}$

Enf_e: $\dfrac{\Gamma \vdash \Delta}{\Gamma,\alpha \vdash \Delta}$

Enf_d: $\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \alpha,\Delta}$

Int_e: $\dfrac{\Gamma,\alpha,\beta,\Delta \vdash \Theta}{\Gamma,\beta,\alpha,\Delta \vdash \Theta}$

Int_d: $\dfrac{\Gamma \vdash \alpha,\beta,\Delta,\Theta}{\Gamma \vdash \beta,\alpha,\Delta,\Theta}$

Con_e: $\dfrac{\Gamma,\alpha,\alpha \vdash \Delta}{\Gamma,\alpha \vdash \Delta}$

Con_d: $\dfrac{\Gamma \vdash \alpha,\alpha,\Delta}{\Gamma \vdash \alpha,\Delta}$

No cálculo de sequentes, também existem regras para os operadores lógicos, chamadas regras lógicas. Para cada um dos operadores lógicos teremos uma regra para introduzir o conectivo no lado esquerdo e outra para o lado direito.

Regras Lógicas:

$\wedge_d$: $\dfrac{\Gamma \vdash \alpha,\Delta \quad \Gamma \vdash \beta,\Delta}{\Gamma \vdash \alpha \wedge \beta,\Delta}$

$\wedge_e$: $\dfrac{\Gamma,\alpha,\beta \vdash \Delta}{\Gamma,\alpha \wedge \beta \vdash \Delta}$

$\vee_d$: $\dfrac{\Gamma \vdash \alpha,\beta,\Delta}{\Gamma \vdash \alpha \vee \beta,\Delta}$

$\vee_e$: $\dfrac{\Gamma,\alpha \vdash \Delta \quad \Gamma,\beta \vdash \Delta}{\Gamma,\alpha \vee \beta \vdash \Delta}$

$\rightarrow_d$: $\dfrac{\Gamma,\alpha \vdash ,\beta,\Delta}{\Gamma \vdash \alpha \rightarrow \beta,\Delta}$

$\rightarrow_e$: $\dfrac{\Gamma \vdash \alpha,\Delta \quad \Gamma,\beta \vdash \Delta}{\Gamma,\alpha \rightarrow \beta \vdash \Delta}$

$\neg_d$: $\dfrac{\Gamma,\alpha \vdash \Delta}{\Gamma \vdash \neg\alpha,\Delta}$

$\neg_e$: $\dfrac{\Gamma \vdash \alpha,\Delta}{\Gamma,\neg\alpha \vdash \Delta}$

Exemplo: Uma dedução no cálculo de sequentes para $\vdash \neg(\alpha \wedge \beta) \rightarrow (\neg\alpha \vee \neg\beta)$ é dada pela seguinte árvore de dedução.

Inc: $\dfrac{}{\alpha \vdash \alpha}$

Enf_e: $\dfrac{}{\alpha,\beta \vdash \alpha}$ Inc: $\dfrac{}{\beta \vdash \beta}$

Int_e: $\dfrac{}{\beta,\alpha \vdash \alpha}$ Enf_e: $\dfrac{}{\beta,\alpha \vdash \beta}$

$\wedge_d$: $\dfrac{}{\beta,\alpha \vdash \alpha \wedge \beta}$

$\neg_d$: $\dfrac{}{\beta \vdash \neg\alpha,\alpha \wedge \beta}$

Int_d: $\dfrac{}{\beta \vdash \alpha \wedge \beta,\neg\alpha}$

$\neg_d$: $\dfrac{}{\vdash \neg\alpha, \alpha \wedge \beta, \neg\beta}$

Int_d: $\dfrac{}{\vdash \alpha \wedge \beta, \neg\beta, \neg\alpha}$

$\neg_e$: $\dfrac{}{\neg(\alpha \wedge \beta) \vdash \neg\beta, \neg\alpha}$

Int_d: $\dfrac{}{\neg(\alpha \wedge \beta) \vdash \neg\alpha, \neg\beta}$

$\vee_d$: $\dfrac{}{\neg(\alpha \wedge \beta) \vdash \neg\alpha \vee \neg\beta}$

$\rightarrow_d$: $\dfrac{}{\vdash \neg(\alpha \wedge \beta) \rightarrow (\neg\alpha \vee \neg\beta)}$

Lógica de Predicados

A linguagem da Lógica de Predicados (LP) ou Linguagem de Primeira Ordem é mais rica que a da Lógica Proposicional, pois além de conter os objetos desta, a linguagem da Lógica de Predicados contém quantificadores, símbolos funcionais e de predicados. Neste sentido, a Lógica de Predicados é uma extensão da Lógica Proposicional. (SOUZA, 2002).

Como na linguagem da Lógica Proposicional é definido inicialmente o alfabeto e em seguida o os outros elementos da linguagem.

O alfabeto da linguagem da LP também é uma extensão do alfabeto da Lógica Proposicional, que é constituído por:

- $X = \{x, y, z, x_1, x_2, ..., y_1, y_2, ..., z_1, z_2, ...\}$ é um conjunto contável de variáveis.
- $\Sigma_C = \{a, b, c, a_1, a_2, ..., b_1, b_2, ..., c_1, c_2, ...$ verdadeiro, falso$\}$ é um conjunto contável de símbolos constantes.
- $\Sigma_F = \{f_1, f_2, f_3, ...\}$ é um conjunto de símbolos de funções não lógicas.
- $\Sigma_R = \{R_1, R_2, R_3, ...\}$ é um conjunto de símbolos de relação ou predicado.
- $\Sigma_L = \{\neg, \vee, \wedge, \rightarrow, \leftrightarrow, \forall, \exists\}$ é o conjunto de símbolos lógicos.
- $\Sigma_P = \{(,), ., ,\}$ é o conjunto de símbolos de pontuação.

Os símbolos para variáveis e as funções formam um novo conjunto que não ocorre na Lógica Proposicional.

O quantificador $\exists$ é chamado de quantificador existencial, pois pode ser lido como “existe um” ou “para algum”. Já o quantificador $\forall$ é chamado de quantificador universal, pois pode ser lido como “para todo” ou “para cada”.

A linguagem dos termos (L^T) da Lógica dos Predicados é gerada por:

T_1: $\frac{-}{x}$, $x \in X$

T_2: $\frac{-}{a}$, $a \in \Sigma_C$

T_3: $\frac{t_1,...,t_n}{f(t_1,...,t_n)}$, $t_j \in L^T$ para todo $1 \leq j \leq n$

As fórmulas da LP são geradas por $\{\sum, G\}$, onde G é a gramática constituída pelas seguintes regras:

F_1: $\frac{t_1,\ldots,t_n}{R(t_1,\ldots,t_n)}$, $t_j \in L^T$ para todo $1 \leq j \leq n$

F_2: $\frac{\alpha}{\neg\alpha}$

F_3: $\frac{\alpha,\beta}{\alpha \wedge \beta}$

F_4: $\frac{\alpha,\beta}{\alpha \vee \beta}$

F_5: $\frac{\alpha,\beta}{\alpha \rightarrow \beta}$

F_6: $\frac{\alpha,\beta}{\alpha \leftrightarrow \beta}$

F_7: $\frac{\alpha}{\exists x.\alpha}$, $x \in X$

F_8: $\frac{\alpha}{\forall x.\alpha}$, $x \in X$

α e β são meta variáveis, isto é, podem ser substituídas nas regras, por qualquer fórmula da LP. (BEDREGAL; ACIÓLY, 2007).

As fórmulas geradas por F_1 são chamadas de fórmulas atômicas, pois elas não contêm nenhum operador lógico.

Como na Lógica Proposicional, a ordem de precedência dos conectivos é utilizada para simplificar as fórmulas, retirando símbolos de pontuação.

A ordem de precedência na Lógica dos predicados é a seguinte:

- Maior precedência: ¬.
- Precedência intermediária superior: ∀, ∃.
- Precedência intermediária inferior: →, ↔.
- Precedência inferior: ∧, ∨.

Exemplos:

¬p(x) ∨ R

((∃y)¬q(y)) ∧ r(z)

((∀x)(p(x) → R))

Semântica

As fórmulas da Lógica de Predicados contêm símbolos que não ocorrem nas fórmulas da L_P. Assim, o significado semântico destas fórmulas é obtido de uma forma diferente da que foi considerada na Lógica Proposicional. (SOUZA, 2002).

Seja U um conjunto não vazio. Uma interpretação I sobre o domínio U, na lógica de predicados é uma função tal que:

- O domínio da função I é o conjunto dos símbolos de função, de predicados e das expressões da LP.
- Para toda variável x, se $I[x] = x_I$, então $x_I \in U$.

- Para todo símbolo de função f, n-ário, se $I[f] = f_I$, então f_I é uma função n-ária, isto é, $f_I: U^n \rightarrow U$.

- Para todo símbolo de predicado p, n-ário, se $I[p] = p_I$, então p_I é um predicado n-ário em U, isto é, $p_I: U^n \rightarrow \{V, F\}$.

A seguir, são consideradas as regras semânticas básicas que definem a interpretação de fórmulas sem quantificadores no seu início. Estas regras definem a interpretação de expressões da lógica de predicado, que não possuem quantificadores no seu início. Analogamente ao que ocorre na L_P, a definição de interpretação das expressões é feita, inicialmente, a partir da definição da interpretação dos símbolos do alfabeto. Em seguida as regras semânticas determinam procedimentos para a interpretação de fórmulas, a partir dos elementos que as constituem.

- Se E = falso, então I[E] = I[falso] = F.

- Se $E = f(t_1, ..., t_n)$ onde $f(t_1, ..., t_n)$ é um termo, então $I[E] = I[f(t_1, ..., t_{nI})] = f(t_{1I}, ..., t_{nI})$ onde $I[f] = f_I$ e para todo termo t_i, $I[t_i] = t_{iI}$.

- Se $E = p(t_1, ..., t_n)$ onde $p(t_1, ..., t_n)$ é um átomo, então $I[E] = I[p(t_1, ..., t_n)] = p_I(t_{1I}, ..., t_{nI})$ onde $I[p] = p_I$ e para todo termo t_i, $I[t_i] = t_{iI}$.

- Se $E = \neg H = V$ se $I[H] = F$ e $I[E] = I[\neg H] = F$ se $I[H] = V$.

- Se $E = H \vee G$, onde H e G são fórmulas, então $I[E] = I[H \vee G] = V$ se $I[H] = V$ e/ou $I[G] = V$ e $I[H \vee G] = F$ se $I[H] = I[G] = F$.

Considerando H uma fórmula e x uma variável.

- $I[(\forall x)H] = V \Leftrightarrow$ "$\forall$" $d \in U$, $<x \leftarrow d>$ $I[H] = V$.

- $I[(\forall x)H] = F \Leftrightarrow$ "$\exists$" $d \in U$, $<x \leftarrow d>$ $I[H] = F$.

- $I[(\exists x)H] = V \Leftrightarrow$ "$\exists$" $d \in U$, $<x \leftarrow d>$ $I[H] = V$.

- $I[(\exists x)H] = F \Leftrightarrow$ "$\forall$" $d \in U$, $<x \leftarrow d>$ $I[H] = F$.

Exemplo: Seja I uma interpretação sobre o domínio U dos números racionais $\mathbb{Q}^*$, diferentes de zero, tal que

$I[a] = 1$, $I[b] = 25$, $I[x] = 13$, $I[y] = 77$, $I[f] = \div$, $I[p] = <$.

Considere a fórmula $G = (\forall x)(\exists y)p(x,y) \rightarrow p(b, f(a,b))$.

Para mostrar que $I[G] = V$. Suponha por absurdo que $I[G] = F$.

Se $I[G] = F$ então $I[(\forall x)(\exists y)p(x,y) \rightarrow p(b, f(a,b))] = F$ e

$I[(\forall x)(\exists y)p(x,y)] = V$ e $I[p(b, f(a,b))] = F$

Observe que $I[p(b, f(a,b))]$ equivale a afirmação $(25 < (1 \div 25))$, que é falsa. Porém

$I[(\forall x)(\exists y)p(x,y)] = V \Leftrightarrow \forall\ d \in \mathbb{Q}^*, <x \leftarrow d>\ I[(\exists y)p(x,y)] = V$

$\Leftrightarrow \forall\ d \in \mathbb{Q}^*, \exists\ c \in \mathbb{Q}^*, <y \leftarrow c><x \leftarrow d>\ I[p(x,y)] = V$

$\Leftrightarrow \forall\ d \in \mathbb{Q}^*, \exists\ c \in \mathbb{Q}^*;\ d < c$ é verdadeiro

Como a última afirmação é verdadeira, então $I[(\forall x)(\exists y)p(x,y)] = V$.

Mas, como $I[(\forall x)(\exists y)p(x,y)] = V$ e $I[p(b, f(a,b))] = F$, conclui-se que $I[G] = F$.

Apresentaremos a seguir os métodos de prova mais adequados para implementação computacional visto anteriormente na Lógica Proposicional, agora na Lógica de Predicados.

Tableaux Semânticos

Os tableaux semânticos na Lógica de Predicados segundo (SOUZA, 2002), utilizam a linguagem da Lógica de Predicados, sendo uma extensão do tableau semântico da Lógica Proposicional.

Um tableau semântico na Lógica de Predicados é uma sequencia de fórmulas, que se apresenta sob a forma de uma árvore, construída de forma análoga aos tableaux da Lógica Proposicional.

Os elementos básicos de um tableau semântico, na Lógica de Predicados, são definidos pela composição dos elementos:

- O alfabeto da Lógica de Predicados.
- O conjunto das fórmulas da Lógica dos Predicados.
- Um conjunto de regras de dedução.

As regras de inferência do tableau semântico na Lógica de Predicados são de R_1 a R_{13}. As regras de R_1 a R_9 são as mesmas utilizadas no tableau semântico na Lógica Proposicional. Vejamos então as regras R_{10}, ..., R_{13}.

$R_{10} = \dfrac{\neg(\forall x)A}{(\exists x)\neg A}$ $\qquad$ $R_{12} = \dfrac{(\exists x)A}{A(t)}$ onde t é novo

$R_{11} = \dfrac{\neg(\exists x)A}{(\forall x)\neg A}$ $\qquad$ $R_{13} = \dfrac{(\forall x)A}{A(t)}$ onde t é qualquer.

As regras R_{10} e R_{11} trocam as posições da negação ¬ dos quantificadores. Na regra R_{10}, tendo $\neg(\forall x)A$ é deduzido $(\exists x)\neg A$, o quantificador universal é transformado em um quantificador existencial. Analogamente, em R_{11} o quantificador existencial é transformado em um quantificador universal.

A regra R_{12}, tendo $(\exists x)$A é deduzido A(t), onde t é um termo novo, que ainda não apareceu na prova. Na regra R_{13}, tendo $(\forall x)$A é deduzido A(t), onde t é um termo qualquer.

A construção de um tableau semântico na Lógica de Predicados é análoga à construção na Lógica Proposicional, assim como os conceitos de tableau fechado e aberto e a forma de provar.

Exemplo:

Seja H = (∀x)(Bom(x) → Alegria) → (∃x)(Bom(x) → Alegria).

A construção do tableau sobre ¬H é dado por:

.1	¬((∀x)(Bom(x) → Alegria) → (∃x)(Bom(x) → Alegria))	¬H
.2	(∀x)(Bom(x) → Alegria)	R_8, .1
.3	¬(∃x)(Bom(x) → Alegria))	R_8, .1
.4	(∀x)¬(Bom(x) → Alegria)	R_{11}, .3
.5	(∀x)Bom(x)	R_8, .4
.6	¬Alegria	R_8, .4
.7	Bom(a)	R_{13}, .5, faça t = a.
.8	(∀x)¬Bom(x) ↙ ↘ $\text{Alegria}_{\text{fech.}}$	R_3, .2
.9	$\neg\text{Bom}(a)_{\text{fech.}}$	R_{13}, .8, faça t = a.

O tableau é fechado, todos os seus ramos são fechados. Logo H é uma tautologia.

Exemplo:

Seja G = ((∀x)(p(x) → q(x)) → ((∃x)p(x) → (∀x)q(x))).

A construção do tableau sobre ¬G é dado por:

.1	¬((∀x)(p(x) → q(x)) → ((∃x)p(x) → (∀x)q(x)))	¬G
.2	(∀x)(p(x) → q(x))	R_8, .1
.3	¬((∃x)p(x) → (∀x)q(x))	R_8, .1
.4	(∃x)p(x)	R_8, .3
.5	¬(∀x)q(x))	R_8, .3
.6	(∃x)¬q(x)	R_{10}, .5
.7	p(a)	R_{12}, .4, faça t = a
.8	¬q(b)	R_{12}, .6, faça t = b
.9	p(a) → q(a)	R_{13}, .2, faça t = a

↙ ↘

.10 $\neg p(a)_{fech.}$ $q(a)_{aberto}$ R_3, .9

O tableau não é fechado, pois contem um ramo aberto. Logo G não é uma tautologia.

Resolução

A resolução na Lógica de Predicados segundo (SOUZA, 2002) é uma extensão da resolução a Lógica Proposicional. Esta extensão ocorre devido à utilização da linguagem da Lógica de Predicados. O método da resolução na LP considera os mesmos conceitos: cláusulas, literal, regra de resolução. Entretanto, devido à presença de variáveis e símbolos de função algumas modificações devem ser consideradas.

Os elementos básicos da resolução na Lógica de Predicados são análogos àqueles da resolução da Lógica Proposicional.

- O alfabeto da Lógica de Predicados.
- O conjunto das cláusulas da Lógica de Predicados.
- A regra de resolução na Lógica dos predicados.

O mecanismo de inferência da resolução utiliza apenas a regra de dedução.

A expansão por resolução na Lógica de Predicados também é análoga ao mesmo conceito na Lógica Proposicional.

Na Lógica Proposicional, toda fórmula possui uma forma clausal associada equivalente. Na Lógica dos Predicados a forma clausal Hc não necessariamente equivale a H. Tem-se apenas que H é insatisfatível se e somente se Hc é insatisfatível. Para determinar Hc a partir de H, incialmente é determinada uma forma prenex Hp associada a H. Em seguida, é considerada a skolemização de Hp obtendo uma fórmula do tipo $(\forall^*)G$ sem quantificadores existenciais tal que G é aberta. O passo final é a transformação de G em uma conjunção de disjunções de literais. No passo final os quantificadores universais são omitidos e a fórmula é escrita na notação de conjuntos, assim como na LP.

Exemplo:

Seja $H = (\forall x)p(x) \wedge (\forall x)(\forall z)((\forall x)q(x) \rightarrow (\exists y)r(x, y, z))$.

Cuja forma prenex é:

$Hp = (\forall y_1)(\forall x)(\forall z)(\exists y_2)(\exists y_3)(p(y_1) \wedge (\neg q(y_2) \vee r(x, y_3, z))$.

A skolemização de Hp tem como resultado:

$Hs = (\forall y_1)(\forall x)(\forall z)(p(y_1) \wedge (\neg q(f(y_1, x, z)) \vee r(x,g(y_1, x, z),z))$.

Representando essa clausula na forma de conjunto, temos que

$Hc = \{\{p(y_1)\}, \{\neg q(f(y_1, x, z)), r(x, g(y_1, x, z), z))\}\}$

Seja H uma formula e ¬Hc a forma clausal associada a ¬H. Uma prova de H por resolução é uma expansão por resolução fechada sobre ¬Hc. Neste caso, H é um teorema do sistema de resolução. Além disso, diz-se que a expansão por resolução associada a ¬Hc é um também uma expansão associada a H.

Exemplo retirado de (SOUZA, 2002, p. 267). Considere o argumento.

Toda pessoa é sábia ou é politiqueira.

Zé não é politiqueiro.

Portanto, Zé é sábio.

É apresentada a seguir, utilizando resolução, uma prova dos argumentos acima. Seja I uma interpretação sobre o domínio U das pessoas, tal que:

$I[q(x)] = V \Leftrightarrow x_I$ é sábio,

$I[p(x)] = V \Leftrightarrow x_I$ é politiqueiro,

I[a] = Zé.

Utilizando esta interpretação, a representação destes argumentos na LP é:

$H = ((\forall x)(p(x) \vee q(x)) \wedge \neg p(a)) \rightarrow q(a)$

Logo,

$\neg H = \neg(((\forall x)(p(x) \vee q(x)) \wedge \neg p(a)) \rightarrow q(a))$ equivale a

$\neg(\neg((\forall x)(p(x) \vee q(x)) \wedge \neg p(a)) \vee q(a))$ equivale a

$((\forall x)(p(x) \vee q(x)) \wedge \neg p(a)) \wedge \neg q(a))$

$\neg Hp = (\forall x)((p(x) \vee q(x)) \wedge \neg p(a)) \wedge \neg q(a))$, que está na forma prenex e não possui quantificadores existenciais. ¬Hp é escrita na notação de conjuntos como:

$Hc = \{\{p(x), q(x)\}, \{\neg p(a)\}, \{\neg q(a)\}\}$

Uma expansão por resolução fechada sobre esse argumento, que é associada a H, é:

.1	$\{p(x), q(x)\}$	
.2	$\{\neg p(a)\}$	
.3	$\{\neg q(a)\}$	
.4	$\{q(x)\}$	Res(.1, .2), $\theta_1 = \{x \leftarrow a\}$.
.5	$\{\ \}$	Res(.3, .4), $\theta_2 = \{x \leftarrow a\}$.

Sistema de Gentzen

Como a Lógica de Predicados é uma extensão da L_P, só incorporaremos as respectivas regras lógicas para os quantificadores universal e existencial, assim como uma restrição ao uso da regra $\rightarrow_d$ e à regra $\neg_d$, que é equivalente a aplicar o teorema da dedução.

Regras Lógicas

$$\forall_e: \frac{\Gamma,\alpha\left[\frac{t}{x}\right] \vdash \Delta}{\Gamma,\forall x.\alpha \vdash \Delta} \qquad \forall_d: \frac{\Gamma \vdash \alpha\left[\frac{t}{x}\right],\Delta}{\Gamma \vdash \forall x.a,\Delta}$$

$$\exists_e: \frac{\Gamma,\alpha\left[\frac{a}{x}\right] \vdash \Delta}{\Gamma,\exists x.\alpha \vdash \Delta} \qquad \exists_d: \frac{\Gamma \vdash \alpha\left[\frac{t}{x}\right],\Delta}{\Gamma \vdash \exists x.\alpha,\Delta}$$

Em todos os casos, t é um termo livre para x em α. Em ∃e, a constante a não deve ocorrer em Γ, Δ nem α. As restrições impostas ao uso do teorema da dedução aqui também são necessárias, uma vez que a regra $\rightarrow_d$ é uma versão de sequentes deste teorema e a regra $\neg_d$ incorpora implicitamente este teorema. Logo, devemos restringir o uso desta regra: a regra $\rightarrow_d$ só poderá ser aplicada, no cálculo de sequentes de Gentzen da lógica de predicados, quando no ramo que deriva $\Gamma,\alpha \vdash \beta,\Delta$ não se aplicou nenhuma regra $\forall_d$ ou $\exists_e$, introduzindo uma variável x que é livre em α. (BEDREGAL; ACIÓLY, 2007).

A seguir daremos uma prova do sequente $\forall x.\alpha \vdash \exists x.\alpha$.

Inc: $\dfrac{}{\alpha \vdash \alpha}$

$\forall_e$: $\dfrac{}{\forall x.\alpha \vdash \alpha}$

$\exists_d$: $\dfrac{}{\forall x.\alpha \vdash \exists x.\alpha}$

Considerando que x é livre para x em α, para qualquer variável x e qualquer fórmula α, temos que α[x/x] é α. Assim, as aplicações das regras $\forall_e$ e $\exists_d$ nesta prova são corretas.

Exemplo: Uma dedução no cálculo de sequentes para $\exists x(Px \wedge Qx) \vdash \exists xPx \wedge \exists xQx$ é dada pela seguinte árvore de dedução.

$$\wedge_d: \dfrac{\text{Enf.: } \dfrac{Pb \vdash Pb}{Pb, Qb \vdash Pb} \qquad \exists_d: \dfrac{\dfrac{Qb \vdash Qb}{Pb, Qb \vdash Qb}\ \text{enf.}}{Pb, Qb \vdash \exists xQx}}{Pb, Qb \vdash Pb \wedge \exists xQx}$$

$$\exists_d: \frac{}{Pb, Qb \vdash \exists xPx \wedge \exists xQx}$$

$$\wedge_e: \frac{}{Pb \wedge Qb \vdash \exists xPx \wedge \exists xQx}$$

$$\exists_e: \frac{}{\exists x(Px \wedge Qx) \vdash \exists xPx \wedge \exists xQx}$$

$$\rightarrow_d: \frac{}{\vdash \exists x(Px \wedge Qx) \rightarrow \exists xPx \wedge \exists xQx}$$

CONSIDERAÇÕES FINAIS

Com esse trabalho, fornecemos um embasamento teórico e prático de alguns métodos de prova em Lógica, com uma linguagem acessível para o aluno de graduação.

É importante destacar que os métodos apresentados não são os únicos, pois existem outros métodos de prova, além dos que foram citados e que não há métodos específicos de prova para determinado argumento ou teorema. Uma prova pode ser feita por vários métodos. É necessário amadurecimento para saber qual método seria mais adequado e menos custoso.

Este trabalho mostra que Lógica é uma disciplina vasta, com muitas aplicações e que não se restringem a aplicações em disciplinas de Matemática. A principal conclusão a que se chega é que podemos raciocinar algoritmamente, ou seja, utilizando passos pré-definidos e seguros para chegarmos a um resultado. Por isso não se concebe nos dias de hoje um mundo sem utilização de computadores para realizar tarefas que podem ser planejadas e programadas.

REFERÊNCIAS

SOUZA, J. N. Lógica para Ciência da Computação: Fundamentos de linguagem, semântica e sistemas de dedução. Rio de Janeiro: Campus, 2002.

MAIA, E. C. S. Notas de Aulas da Disciplina Introdução a Lógica Matemática. Vitória da Conquista. 2005. (Apostila).

BEDREGAL, B. R. C.; ACIÓLY, B. M. Introdução à Lógica Clássica para a Ciência da Computação. Natal. 2007. (Apostila). Disponível em <www.dimap.ufrn.br/~jmarcos/books/BA_Jul07.pdf>. Acesso em 05 Abril 2015.

PEREIRA, S. L. Lógica Proposicional. São Paulo. 2011. (Apostila). Disponível em <http://correio.fdvmg.edu.br/downloads/DET147/IA-logicaProposicional.pdf> acesso em 02 de Fevereiro 2016.