

Raciocínio por Tablôs pela Forma Direta

Arthur Buchsbaum¹, Maurício Correia Lemes Neto²

¹ Departamento de Informática e Estatística, Universidade Federal de Santa Catarina
Campus Universitário, Trindade, Florianópolis, SC, Brasil
arthur@inf.ufsc.br

² Coordenação de Informática, Universidade Tecnológica Federal do Paraná
Av. Alberto Carazzai nº 1640, Campus Cornélio Procópio, Cornélio Procópio, PR, Brasil
mauricio@cp.cefetpr.br

Resumo – Este artigo apresenta uma forma alternativa de geração de árvores de prova por tablôs. Denominamos esse método de *direto*, por causa da característica em que a possível conclusão é inserida no tablô inicial sem negá-la. Já o método dos tablôs por *refutação* se utiliza da negação da possível conclusão. No sistema de tablôs por prova direta para a lógica clássica, cada ramo corresponde semanticamente à disjunção das fórmulas que o compõem, e cada tablô equivale semanticamente à conjunção de todas essas disjunções. Em qualquer um dos métodos baseados em tablôs para a Lógica Clássica, tanto *direto* quanto *indireto*, um ramo é considerado fechado se o mesmo contiver duas fórmulas contraditórias. No método direto o fechamento de um ramo acarreta a sua validade semântica, a qual por sua vez implica, no caso do fechamento de todos os ramos, na validade da possível conclusão. Já no método indireto o fechamento de todos os ramos acarreta a insatisfatibilidade da negação da possível conclusão, o que por sua vez implica na validade da mesma.

Palavras-chave: Tablô, Sistema de Tablôs, Método Direto, Método Indireto, Método da Refutação.

Abstract – This article presents an alternative form of generating tree proofs by tableaux, which is denominated *direct*, due to the characteristic in which the possible conclusion is inserted in the initial tableau, without denying it. The tableau method by *refutation* uses instead the denial of the possible conclusion. In the tableau system by direct proof for classical logic, each branch corresponds semantically to the disjunction of all formulas that compose it, and any tableau is semantically equivalent to the conjunction of all these disjunctions. For each one of the tableau based methods for classical logic, the *direct* or the *indirect* one, a branch is considered closed if the same contains two contradictory formulas. In the direct method the closure of a branch corresponds to its validity, which implies, in case of closure of all branches, to the validity of the possible conclusion, whereas in the indirect method the closure of all branches entails the unsatisfiability of the denial of the possible conclusion, which in its turn implies its validity.

Keywords: Tableau, Tableau System, Direct Method, Indirect Method, Refutation Method.

Introdução

O conceito de inteligência é altamente complexo para ser bem definido, sendo que as diversas tentativas de explicá-lo são em geral incompletas. Em particular, a inteligência artificial (IA) possui diversas interpretações entre a comunidade científica, muitas delas bem imprecisas e cheias de lacunas. Esta área de pesquisa tem apresentado um crescimento notável nas últimas décadas e várias técnicas para a construção de algum tipo de inteligência foram desenvolvidas, apesar de serem ainda restritas a uma pequena parcela da realidade. Dentre essas técnicas está a automatização do raciocínio, que utiliza a lógica na construção de sistemas aptos a simular algumas formas de inteligência.

A lógica simbólica dedica-se ao estudo de sistemas formais, através de métodos matemáticos, que se integram em uma unidade, uma coleção de formas de raciocínio, aptas a

interagir entre si de uma forma conjugada, sendo que a maioria desses sistemas atende a diversas demandas de possíveis aplicações. A ênfase do trabalho completo [1] deu-se nos sistemas relacionados à Lógica Clássica, nos níveis proposicional e quantificacional, pois esta é uma base de acesso à compreensão de vários dos sistemas lógicos mais importantes.

Entre os métodos de automatização do raciocínio podemos citar o *método da resolução* [2,3], o *método dos seqüentes de Gentzen* [4,5], e o *método dos tablôs* [5,6,7].

O primeiro método, o da resolução, perfaz uma complexa manipulação formal inicial das fórmulas de entrada. O seu algoritmo principia com vários procedimentos destinados a reduzir as fórmulas dadas inicialmente a outras fórmulas em

forma *prenex*¹, eliminando-se a seguir todos os quantificadores existenciais através de um método dito *skolemização*, bem como colocando na forma disjuntiva normal as subfórmulas livres de quantificadores.

O segundo método envolve a manipulação formal de certos objetos sintáticos chamados de *seqüentes*, os quais compreendem uma lista de fórmulas, seguida de um sinal separador, sucedida por uma outra lista de fórmulas.

Assim sendo, tanto o método da resolução como o método dos seqüentes não primam pela simplicidade, pois o primeiro envolve uma complexa manipulação simbólica antes do trabalho central do seu algoritmo correspondente, e o segundo não lida diretamente com fórmulas como objetos sintáticos básicos, mas sim com seqüentes.

Já o método dos tablôs destaca-se pela sua inerente simplicidade, pois, ao contrário do método dos seqüentes, lida diretamente com fórmulas, e, ao contrário do método da resolução, não exige complexos procedimentos iniciais de preparação das fórmulas envolvidas para que a parte central do seu algoritmo correspondente seja colocada em funcionamento.

Neste trabalho mostra-se que, no método dos tablôs, é possível proceder de uma outra forma, envolvendo uma análise semântica direta da conclusão, sem qualquer refutação, e sim através de uma justificativa direta da validade da conclusão.

O Método dos Tablôs

Um tablô é uma árvore de fórmulas em uma dada lógica. A partir de uma fórmula inicial, a qual corresponde a uma possível tese de uma dada lógica ou com a sua negação, é gerada uma árvore ou *tablô inicial*.

Esta árvore é a primeira de uma seqüência de árvores, onde cada árvore não inicial sucede a anterior através da aplicação de uma *regra de expansão* a um nó não usado ou marcado. Cada ramo desta árvore em evolução pode estar aberto ou fechado, segundo um *critério de fechamento* previamente dado. Ramos fechados não crescem mais, enquanto que os abertos são expandidos através da aplicação de regras aos seus nós ainda não usados. Este processo pára quando for encontrado um tablô com todos os ramos fechados, ou quando for encontrado um ramo aberto em que todos os nós não usados não forem aplicáveis a nenhuma regra. No primeiro caso, se for constatada a correção do algoritmo com respeito à lógica considerada, então a possível tese é válida, enquanto que, no segundo caso, se

for confirmada a completude do algoritmo, então a possível tese não é válida.

Expomos aqui, em contraste com a forma indireta ou da refutação, um método dos tablôs por prova direta. O método dos tablôs por refutação, cujos principais precursores foram Beth [8] e Hintikka [9], é um processo algorítmico capaz de provar um dado teorema pelo insucesso de uma busca exaustiva de uma interpretação que satisfaça a sua negação. De forma semelhante, o método dos tablôs por prova direta também é um processo algorítmico, mas sua principal diferença, em relação ao método indireto, está em provar uma dada tese de forma direta, ou seja, sem que seja necessário negá-la.

O método dos tablôs direto também consiste em desenvolver uma árvore de fórmulas (tablô) a partir de um tablô inicial, porém o mesmo contém a fórmula correspondente a uma possível tese, ao invés da negação da mesma.

Na próxima seção definimos o que vem a ser um sistema de tablôs, de uma forma independente de um dado sistema lógico particular.

Sistemas de Tablôs

Consideramos duas lógicas, L e L' respectivamente as suas linguagens, denominadas respectivamente de *linguagem inicial* e *linguagem de trabalho*. A primeira lógica é aquela para a qual se objetiva construir um sistema de tablôs, enquanto que a segunda é auxiliar, isto é, a mesma é definida apenas para facilitar, no caso do método dos tablôs, o estudo da lógica principal. O processo algorítmico envolve sempre árvores de fórmulas de L' , isto é, de fórmulas de uma linguagem da lógica auxiliar.

Linguagens de trabalho distintas da linguagem inicial são necessárias em um dos seguintes casos:

- 1) o tratamento de quantificadores exige a inclusão de uma infinidade de constantes à linguagem inicial;
- 2) algumas lógicas não permitem uma análise semântica com o uso exclusivo de sua própria linguagem.

Um tablô em L' é uma árvore de nós cujo conteúdo é, no mínimo, uma fórmula de L' e um dos valores booleanos v ou f . Um nó é dito estar *marcado* se o seu valor booleano for v . Um tablô é dito *finito* se ele possuir um número finito de nós.

Uma *função de inicialização* para L associa uma fórmula de L a um tableau em L' .

Em diversos sistemas de tablôs, a função de inicialização cria um tablô inicial contendo apenas um nó. No caso do método direto tal nó inicial é composto pela provável tese, a menos de pequenas transformações na fórmula inicial. Já no método por refutação o tableau inicial também possui em muitos casos um único nó, que contém

¹ Uma fórmula *prenex* é uma fórmula da forma $\Psi_1 x_1 \dots \Psi_n x_n P$, onde Ψ_1, \dots, Ψ_n são um dos quantificadores " \forall " ou " \exists ", e P não possui quantificadores.

a negação da possível tese, a menos de eventuais transformações.

Um *critério de fechamento* (em L') é uma função que associa ramos de fórmulas de L' a um dos valores booleanos v ou f . Um ramo em L' é dito *fechado* (com respeito a um dado critério de fechamento) se o mesmo for associado a v por tal critério, caso contrário tal ramo é dito *aberto* (neste contexto).

Uma *regra* (em L') é uma função que associa cada fórmula de L' e cada ramo em L' a uma coleção finita de tablôs finitos em L' .

Um *sistema de tablôs* é uma quintupla ordenada $S = \langle L, L', I, C, R \rangle$, onde L e L' são linguagens formais, I é uma função de inicialização para L , C é um critério de fechamento em L' e R é uma coleção de regras em L' . Dizemos também que L é a linguagem inicial de S , L' é a linguagem de trabalho de S , I é a função de inicialização de S , C é o critério de fechamento de S e R é a coleção de regras de S . É dito também que cada elemento de R é uma regra de S .

Os tablôs são estendidos através das regras, ou seja, o crescimento das árvores de fórmulas dá-se pela aplicação de regras às fórmulas contidas em nós não marcados.

Uma *extensão imediata* de um tablô T em S é um tableau T' obtido de T marcando-se um nó escolhido de T ainda não marcado e aplicando-se uma regra de S à fórmula deste nó juntamente com todos os ramos abertos a que este nó pertence, e colocando-se os resultados correspondentes nos finais de todos estes ramos abertos.

Um tablô é dito *fechado* em S se todos os seus ramos forem fechados em S , caso contrário tal tablô é dito ser *aberto* em S .

Uma dada fórmula de L' é dita ser *aplicável* em S se existir uma regra em S tal que qualquer par formado por tal fórmula e um ramo qualquer em L' pertença ao seu domínio; caso contrário tal fórmula é dita ser *excluída* em S .

Um ramo em L' é dito ser *exaurido* em S se o mesmo for aberto em S e todos os seus nós que possuem fórmulas não excluídas em S estiverem marcados. Um tablô em L' é dito ser *exaurido* em S se o mesmo possuir pelo menos um ramo exaurido em S .

Uma *seqüência de desenvolvimento de tablôs* em S é uma seqüência $(T_i)_{i \in I}$ de tablôs em S tal que, para cada $i \in I$, se $i+1 \in I$, então T_{i+1} é uma extensão imediata de T_i em S . Uma tal seqüência é dita ser *completa* em S se a mesma for finita e terminar com um tablô fechado em S ou um tablô exaurido em S , ou se a mesma for infinita e tender para um tablô limite exaurido em S .

O *tableau inicial* para uma fórmula P da linguagem inicial de S é o tablô obtido por uma aplicação da função de inicialização de S a P . Um tablô T' é um desenvolvimento de um tablô T em S se $T = T'$ ou se há uma seqüência de

desenvolvimento de tablôs em S tal que T precede T' nesta seqüência. T é dito ser um tablô para uma fórmula P da linguagem inicial de S se T for um desenvolvimento do tablô inicial para P .

Um sistema de tablôs S é dito ser *correto* com respeito a uma dada lógica L se, dada uma fórmula P em L , houver um tablô fechado para P em S . S é dito ser *completo* com respeito a uma dada lógica L se a existência de um tablô fechado para P em S implicar na validade de P em L .

Um Sistema de Tablôs por Prova Direta para a Lógica Quantificacional Clássica

Definimos a seguir um sistema de tablôs que demonstramos em [1] ser correto e completo com respeito à Lógica Quantificacional Clássica. O mesmo é denominado aqui de **STD** (tal sigla vem de **Sistema de Tablôs pelo Método Direto**). Sistemas de tablôs para algumas lógicas não clássicas foram definidos em [10,11,12].

A linguagem inicial de **STD** é uma linguagem de primeira ordem fixa, tal como é definido em livros-texto de lógica [13,14,15]. A linguagem de trabalho de **STD** é obtida de sua linguagem inicial pelo acréscimo de uma infinidade de novas constantes.

Dada uma fórmula P , obtém-se o tablô inicial para P em **STD** em três etapas:

- 1) obtém-se uma fórmula P' de P pela eliminação de todos os quantificadores vácuos de P , ou seja, cada subfórmula de P de uma das formas $\forall x Q$ ou $\exists x Q$, tal que x não é livre em Q , é substituída por Q ;
- 2) a partir de P' , obtém-se P'' substituindo-se todas as variáveis livres em P' por novas constantes;
- 3) finalmente, forma-se o tablô inicial para P em **STD**, que é um tablô possuindo um único nó, não marcado, cuja fórmula é P'' .

Um ramo é fechado em **STD** se ele possuir duas fórmulas contraditórias.

STD possui treze regras de expansão, as quais lidam com todos os tipos de fórmulas que uma linguagem quantificacional da lógica clássica pode possuir. Todas elas resultam em até dois novos tablôs, que devem ser acrescentados no final de cada ramo aberto ao qual o nó considerado pertença, até ser obtido um tablô fechado ou um tablô exaurido. No primeiro caso conclui-se, levando em conta a correção de **STD**, que a fórmula inicial é válida, ou seja, é uma tese da lógica quantificacional clássica. No segundo caso conclui-se, devido à completude de **STD**, que a mesma não é válida.

Implicação ($P \rightarrow Q$): Esta regra associa cada ramo contendo $P \rightarrow Q$ e a fórmula $P \rightarrow Q$ a um único tablô, com um único ramo contendo dois nós não marcados, cujas fórmulas são respectivamente $\neg P$ e Q . Veja a figura 1.

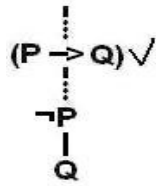


Figura 1 – Regra da Implicação

Conjunção ($P \wedge Q$): Esta regra associa cada ramo contendo $P \wedge Q$ e a fórmula $P \wedge Q$ a dois tablôs, cada um deles possuindo um único ramo contendo um único nó não marcado, cujas fórmulas são respectivamente P e Q . Veja a figura 2.

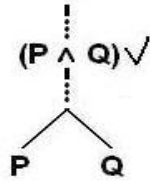


Figura 2 – Regra da Conjunção

Disjunção ($P \vee Q$): Esta regra associa cada ramo contendo $P \vee Q$ e a fórmula $P \vee Q$ a um único tablô com um único ramo contendo dois nós não marcados, cujas fórmulas são respectivamente P e Q . Veja a figura 3.

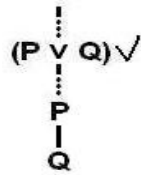


Figura 3 – Regra da Disjunção

Negação de Implicação ($\neg(P \rightarrow Q)$): Esta regra associa cada ramo contendo $\neg(P \rightarrow Q)$ e a fórmula $\neg(P \rightarrow Q)$ a dois tablôs, cada um deles possuindo um único ramo contendo um único nó não marcado, em que a fórmula do primeiro é P e a do segundo é Q . Veja a figura 4.

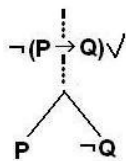


Figura 4 – Regra da Negação de Implicação

Negação de Conjunção ($\neg(P \wedge Q)$): Esta regra associa cada ramo contendo $\neg(P \wedge Q)$ e a fórmula $\neg(P \wedge Q)$ a um único tablô com um único ramo contendo dois nós não marcados, cujas fórmulas são respectivamente $\neg P$ e $\neg Q$. Veja a figura 5.

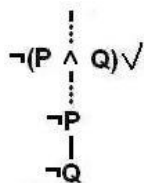


Figura 5 – Regra da Negação de Conjunção

Negação de Disjunção ($\neg(P \vee Q)$): Esta regra associa cada ramo contendo $\neg(P \vee Q)$ e a fórmula $\neg(P \vee Q)$ a dois tablôs, cada um deles possuindo um único ramo contendo um único nó não marcado, em que a fórmula do primeiro é $\neg P$ e a do segundo é $\neg Q$. Veja a figura 6.

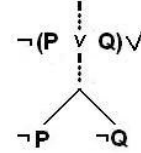


Figura 6 – Regra da Negação de Disjunção

Negação de Negação ($\neg\neg P$): Esta regra associa cada ramo contendo $\neg\neg P$ e a fórmula $\neg\neg P$ a um único tablô com um único nó não marcado cuja fórmula é P . Veja a figura 7.

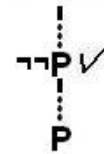


Figura 7 – Regra da Negação de Negação

Equivalência ($P \leftrightarrow Q$): Esta regra associa cada ramo contendo $P \leftrightarrow Q$ e a fórmula $P \leftrightarrow Q$ a dois tablôs, cada um deles possuindo um único ramo contendo dois nós não marcados, o primeiro contendo as fórmulas $\neg P$ e Q , e o segundo contendo as fórmulas P e $\neg Q$. Veja a figura 8.

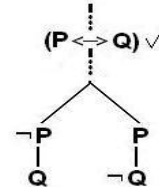


Figura 8 – Regra da Equivalência

Negação de Equivalência ($\neg(P \leftrightarrow Q)$): Esta regra associa cada ramo contendo $\neg(P \leftrightarrow Q)$ e a fórmula $\neg(P \leftrightarrow Q)$ a dois tablôs, cada um deles possuindo um único ramo contendo dois nós não marcados, o primeiro contendo as fórmulas P e Q , e o segundo contendo as fórmulas $\neg P$ e $\neg Q$. Veja a figura 9.

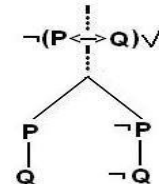


Figura 9 – Regra da Negação de Equivalência

Fórmula Universal ($\forall x P$): Esta regra associa cada ramo contendo $\forall x P$ e a fórmula $\forall x P$ a um único tablô com um único nó não marcado, cuja fórmula é $P(x/c)$, obtida da instanciação da variável x por alguma constante c que ainda não figura no ramo considerado. Veja a figura 10.

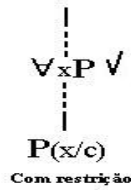


Figura 10 – Regra da Fórmula Universal

Fórmula Existencial ($\exists x P$): Esta regra associa cada ramo contendo $\exists x P$ e a fórmula $\exists x P$ a um tablô com um único ramo com nós não marcados cujas fórmulas são $P(x/t_1), \dots, P(x/t_n)$, onde $P(x/t_1), \dots, P(x/t_n)$ são instanciações de P por todos os termos fechados² que figuram no ramo para os quais P ainda não foi instanciado. Veja a figura 11.

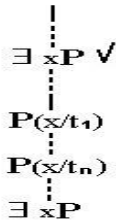


Figura 11 – Regra da Fórmula Existencial

Negação de Fórmula Universal ($\neg(\forall x P)$): Esta regra associa cada ramo contendo $\neg(\forall x P)$ e a fórmula $\neg(\forall x P)$ a um único tablô com um nó não marcado cuja fórmula é $\exists x \neg P$. Veja a figura 12.

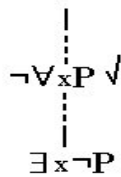


Figura 12 – Regra da Negação de Fórmula Universal

Negação de Fórmula Existencial ($\neg(\exists x P)$): Esta regra associa cada ramo contendo $\neg(\exists x P)$ e a fórmula $\neg(\exists x P)$ a um único tablô com um nó não marcado cuja fórmula é $\forall x \neg P$. Veja a figura 13.

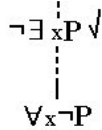


Figura 13 – Regra da Negação de Fórmula Existencial

Exemplo com uma fórmula válida em LQC

Dada a fórmula

$$P \wedge (Q \vee R) \leftrightarrow P \wedge Q \vee (P \wedge R),$$

temos que, conforme a figura 10, há um tablô fechado para esta fórmula em **STD**, e daí ela é uma tese em **LQC**. Esta figura mostra a seqüência em que os tablôs foram desenvolvidos, sendo que, para cada natural n menor ou igual à profundidade desta árvore, o seu numeral correspondente foi

² São termos que não contêm variáveis.

colocado ao lado da representação de cada nó de T_n que não é nó de T_{n-1} . Esta seqüência termina com um tablô fechado em **STD**, e daí esta fórmula é válida em **LQC**.

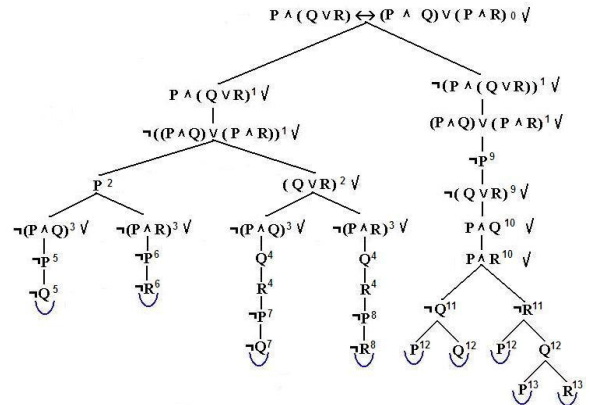


Figura 10 – Exemplo com uma fórmula válida em LQC

Exemplo com uma fórmula inválida em LQC

O tablô para a fórmula

$$P \vee (Q \wedge R) \leftrightarrow (P \vee Q) \vee (P \vee R),$$

exposto na figura 11, possui dois ramos exauridos em **STD**, e daí temos que a fórmula não é válida em **LQC**.

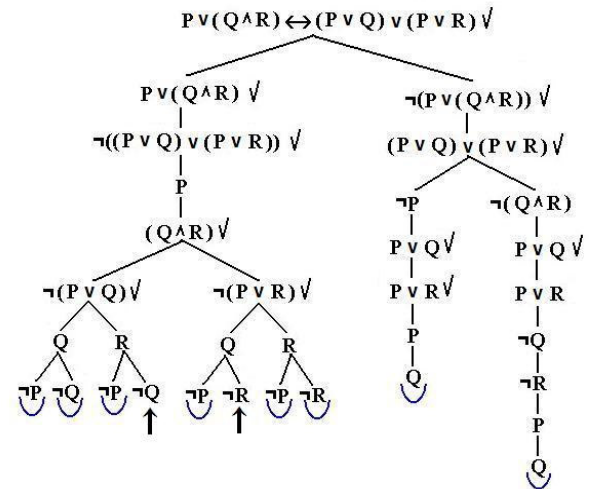


Figura 11 – Exemplo com uma fórmula inválida em LQC

Exemplo com uma fórmula válida contendo quantificadores universais

O desenvolvimento de um tablô por prova direta, exposto na figura 14, para a fórmula

$$\forall x (px \rightarrow qx) \rightarrow (\forall x px \rightarrow \forall x qx),$$

não obedeceu nenhuma ordem pré-definida, porém fez-se o uso de uma ordem mais adequada para obter o fechamento dos ramos mais rapidamente. Assim, obtivemos um tablô fechado para a referida fórmula, portanto temos que $\forall x (px \rightarrow qx) \rightarrow (\forall x px \rightarrow \forall x qx)$ é uma tese da Lógica Quantificacional Clássica.

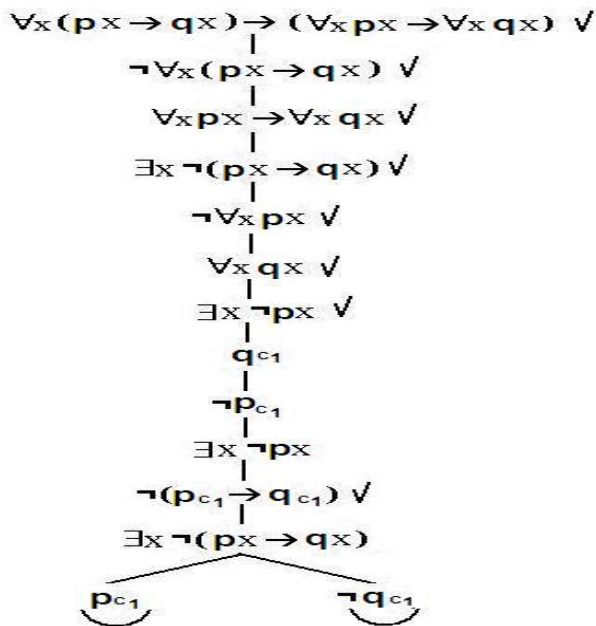


Figura 14 – Exemplo com uma fórmula válida em LQC

Exemplo com uma fórmula inválida contendo quantificadores existenciais

Cada ramo exaurido de um tablô para a fórmula

$$\exists x (px \rightarrow qx) \rightarrow (\exists x px \rightarrow \forall x qx)$$

em STD dá uma interpretação para a qual todas as fórmulas do ramo são falsas e portanto a fórmula inicial não é válida. A figura 15 exibe o tablô final de uma seqüência de desenvolvimento completa em STD, o qual possui dois ramos exauridos abertos. Daí podemos concluir que esta fórmula não é válida em LQC.

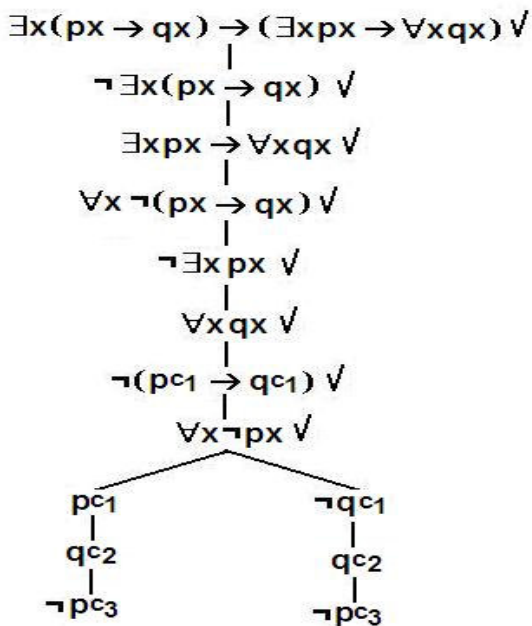


Figura 15 – Exemplo com uma fórmula inválida em LQC

Considerações finais

Definimos neste trabalho sistemas de tablôs para a lógica quantificacional clássica, segundo o método da prova direta. A principal motivação reside no fato de que todas as referências consultadas classificam o método dos tablôs (e o método da resolução) como um procedimento de prova por refutação, o que não é verdadeiro.

Existem condições que um sistema de tablôs por prova direta deve satisfazer para que o mesmo seja correto e completo com respeito a uma lógica munida de uma semântica de valorações, bem como provas de correção e completude levando em conta tais condições. Maiores detalhes estão em [1].

Esperamos que este texto tenha ajudado o leitor a romper com a crença da comunidade científica e acadêmica de que o método dos tablôs é vinculado à refutação, o que vimos que não é necessário.

Referências

1. Lemes Neto, Maurício Correia: *Um Método dos Tablôs por Prova Direta para a Lógica Clássica*. Dissertação de Mestrado em Ciência da Computação, orientação de Arthur Buchsbaum, UFSC, 2004.
2. Chang, Chin-Liang & Lee, Richard Char-Tung.: *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.
3. Loveland, D. W.: *Automated Theorem Proving: A Logical Basis*. North-Holland, 1978.
4. Gentzen, G.: *The Collected Papers of Gerhard Gentzen*. North-Holland, 1969.
5. Smullyan, R.: *First-Order Logic*. Dover, 1995.
6. Buchsbaum, A. & Pequeno, T: *O Método dos Tableaux Generalizado e Sua Aplicação ao Raciocínio Automático em Lógicas Não Clássicas*. Cadernos do Departamento de Filosofia da PUC-Rio, Revista "O que nos faz pensar", 1990, pgs. 81-96.
7. Mortari, C. A.: *Introdução à Lógica*. UNESP, 2001.
8. Beth, E. W. *Semantic Entailment and Formal Derivability*. Mededelingen van de Koninklijke Nederlandse Akademie van Wetenschappen, 18, pgs. 309-342, Amsterdam, 1955.
9. Hintikka, J.: *Form and Content in Quantification Theory*, Acta Philosophica Fennica, 1995.
10. Buchsbaum, A. & Pequeno, T: *A Reasoning Method for a Paraconsistent Logic*. Studia Logica, 52, v. 2, 1993, pgs. 281-289.
11. Buchsbaum, A. & Pequeno, T.: *Automated Deduction with Non Classical Negations*. Proceedings of 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods, 1994, pgs. 51-63.

12. Buchsbaum, A. & Pequeno, T.: *Raciocínio Automático em Lógicas Paraconsistentes e/ou Paracompletas*. 6^o Simpósio Brasileiro em Inteligência Artificial, 1989, pgs. 1-15.
13. Enderton, H. B.: *A Mathematical Introduction to Logic*. Academic Press, 1972.
14. Ebbinghaus, H.-D., Flum, J. & Thomas, W.: *Mathematical Logic*. Springer-Verlag, 1994.
15. Bell, J. & Machover, M.: *A Course in Mathematical Logic*. North-Holland, 1977.