O Método dos Tableaux com Unificação

Arthur Buchsbaum [1]

arthur@inf.ufsc.br

Letícia Carvalho Pivetta Fendt [1,2]

leticia@inf.ufsc.br, leticia@inf.ulbrajp.com.br

[1] Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis / SC – CEP 88040-900
Fone: (0xx48) 331-9738

[2] Curso de Informática
Instituto Luterano de Ensino Superior de Ji-Paraná
Ji-Paraná / RO – CEP: 78 960 000
Caixa Postal: 271
Fone: (0xx69) 416-3144

Abstract

In this work one refinement is proposed on the tableaux method, aiming to decrease the number of nodes of the proof trees, as well as to increase the possibility of obtaining answers. For this a algorithm is specified, based on the traditional tableau method for classical logic, which appeals to the unification procedure, commonly used by the resolution method.

Keywords

Classical Logic, Tableau Method, Automated Reasoning and Unification.

Resumo

No presente trabalho propomos um refinamento sobre o método dos tableaux, tal como tradicionalmente apresentado, visando diminuir o número de nós das árvores de prova, bem como aumentar a possibilidade de obtenção de respostas. Para isto especificamos um algoritmo, baseado no método tradicional dos tableaux para a lógica clássica, o qual recorre ao procedimento de unificação, comumente utilizado no método da resolução.

Palavras chave

Lógica Clássica, Raciocínio Automático, Método dos Tableaux e Unificação.

1 – Introdução

Podemos realizar a simulação de uma dada forma de raciocínio através da escolha e/ou modelagem de uma lógica, e da especificação de um método de prova automático, parcial ou total para tal lógica, os quais melhor reflitam aquela forma de raciocínio.

Um método automático de prova consiste na obtenção de um algoritmo sistemático que decida, da forma mais ampla possível, quando uma dada fórmula é conseqüência lógica de uma dada base de conhecimento. O presente trabalho trata de algoritmos de automatização do processo de raciocínio para a lógica quantificacional clássica, e utiliza, como algoritmo de prova, o método dos tableaux.

A automatização do raciocínio enfrenta dois obstáculos básicos, os quais direcionam a evolução deste campo de pesquisa:

1º) todo algoritmo de automatização, no decorrer do seu processamento, provoca, em geral, um crescimento explosivo do espaço de busca, daí torna-se imperiosa a descoberta de certos refinamentos que diminuam ao máximo a proliferação de dados intermediários desnecessários;

2º) como o problema da prova automática de teoremas é indecidível, todo algoritmo de automatização está sujeito a entrar em um processamento perpétuo para certos dados que deveriam fornecer respostas negativas, daí são necessários também refinamentos que ampliem ao máximo os espaços para os quais obtemos soluções decidíveis.

Atendendo a estas duas demandas, especificamos, neste trabalho, um refinamento para o método dos tableaux, visando assim diminuir ao máximo a proliferação de dados intermediários, bem como otimizar as possibilidades de obtenção de respostas negativas. Tal refinamento faz uso do processo de unificação comumente utilizado no algoritmo de resolução.

2- Sistema de Tableaux Tradicional para Lógica Clássica

O sistema de tableaux, cujos precursores foram Beth e, independentemente, Hintikka [Reeves 1983], é um método de prova por refutação, no qual prova-se um teorema pelo insucesso na tentativa de construção sistemática de um modelo para a sua negação. Originalmente, o método verifica a impossibilidade da satisfação da negação de uma determinada fórmula.

A seguir definiremos o sistema de tableaux tradicional para a lógica quantificacional clássica tal como encontrado habitualmente na literatura existente, como, por exemplo, em [Revees 1983] e [Buchsbaum & Pequeno 1990]. De agora em diante chamaremos o sistema de tableaux que iremos progressivamente especificar de $\mathbf{S_0}$.

A linguagem inicial de $\mathbf{S_0}$ é uma linguagem de primeira ordem fixa, que notamos, de agora em diante, por $\mathbf{L_0}$. A linguagem de trabalho de $\mathbf{S_0}$ é uma extensão de $\mathbf{L_0}$, a qual notamos por $\mathbf{L_1}$, obtida acrescentando-se uma infinidade de novas constantes ao alfabeto de $\mathbf{L_0}$.

A função de inicialização de $\mathbf{S_0}$, a qual notamos por $\mathbf{I_0}$, associa a cada coleção Γ de fórmulas de $\mathbf{L_1}$ um tableau com um único ramo sem fórmulas marcadas, cujas fórmulas são obtidas de Γ substituindo todas as variáveis livres de Γ por novas constantes e retirando os quantificadores vácuos das fórmulas de Γ.

A coleção de regras de $\mathbf{S_0}$, a qual notamos por $\mathbf{R_0}$, é composta das onze regras dadas pela Figura 1.

Regra $\neg\neg P$
$$\begin{array}{c} \neg\neg P \\ | \\ P \end{array}$$

Regra $P \to Q$
$$\begin{array}{cc} \multicolumn{2}{c}{P \to Q} \\ \diagup & \diagdown \\ \neg P & Q \end{array}$$

Regra $P \wedge Q$
$$\begin{array}{c} P \wedge Q \\ | \\ P \\ | \\ Q \end{array}$$

Regra $P \vee Q$
$$\begin{array}{cc} \multicolumn{2}{c}{P \vee Q} \\ \diagup & \diagdown \\ P & Q \end{array}$$

Regra $\exists x\, P$
$$\begin{array}{c} \exists x\, P \\ | \\ P(x/c) \end{array}$$

Regra $\forall x\, P$
$$\begin{array}{c} \forall x\, P \\ | \\ P(x/t_n) \\ \dots \\ P(x/t_1) \\ | \\ \forall x\, P \end{array}$$

Regra $\neg(P \to Q)$
$$\begin{array}{c} \neg(P \to Q) \\ | \\ P \\ | \\ \neg Q \end{array}$$

Regra $\neg(P \wedge Q)$
$$\begin{array}{cc} \multicolumn{2}{c}{\neg(P \wedge Q)} \\ \diagup & \diagdown \\ \neg P & \neg Q \end{array}$$

Regra $\neg(P \vee Q)$
$$\begin{array}{c} \neg(P \vee Q) \\ | \\ \neg P \\ | \\ \neg Q \end{array}$$

Regra $\neg\exists x\, P$
$$\begin{array}{c} \neg\exists x\, P \\ | \\ \forall x\, \neg P \end{array}$$

Regra $\neg\forall x\, P$
$$\begin{array}{c} \neg\forall x\, P \\ | \\ \exists x\, \neg P \end{array}$$

Figura 1: Regras de Expansão dos Ramos para o Sistema de Tableaux Tradicional.

O critério de fechamento de $\mathbf{S_0}$ é uma função, a qual notamos por **F**, que associa a um dado ramo **ρ** em $\mathbf{L_1}$ a palavra **fechado** se **ρ** possui duas fórmulas **P** e ¬**P**’, onde **P** é congruente a **P**’, e a palavra **aberto** caso contrário.

A constante **c**, apresentada na expressão **P**(***x***/**c**) escrita na definição da regra para fórmulas existenciais na Figura 1, é a primeira constante em $\mathbf{L_1}$ que não figura no ramo considerado.

Os termos $\mathbf{t_1}$,..., $\mathbf{t_n}$, apresentados nas expressões $\mathbf{P}(\boldsymbol{x}/\mathbf{t_1})$, …, $\mathbf{P}(\boldsymbol{x}/\mathbf{t_n})$, escritos na definição da regra para fórmulas universais na Figura 1, são todos os termos fechados $\mathbf{t_i}$ (**i =** 1, …,0) do ramo considerado, tais que $\forall \boldsymbol{x}\ \mathbf{P}$ ainda não tenha sido instanciado para $\mathbf{t_i}$.

3 - O Sistema de Tableaux para Lógica Clássica Usando o Método da Unificação

O intuito da construção deste sistema é trazer ao método um pouco mais da intuição humana, que aparece quando o mesmo é executado com papel e lápis. O que qualquer pessoa sensata, que utilize o método dos tableaux, faz, ao pôr o mesmo em prática, é priorizar a expansão de fórmulas que não abrem novos ramos(fórmulas disjuntivas), além de realizar instanciações de fórmulas universais somente com aqueles termos interessantes ao fechamento do ramo em questão.

Nesta modificação, realizada sobre o método dos tableaux, priorizamos a expansão daqueles nós que contém fórmulas conjuntivas e existenciais sobre os demais nós, e incluímos no algoritmo em questão um procedimento de unificação, o qual é uma expansão do algoritmo de unificação usado tradicionalmente nos métodos de resolução, tal como apresentado, por exemplo, em [Lloyd 1987]; outras apresentações deste algoritmo para o método da resolução podem ser encontradas em [Chang & Lee 1973], [Loveland 1978] e [Fitting 1990].

Encontramos em [Fitting 1990] um tipo de associação entre o método dos tableaux e o procedimento de unificação diferente da maneira proposta aqui. Tal método substitui variáveis quantificadas universalmente por variáveis livres, as quais, ao serem substituídas por termos apropriados ao fechamento de determinados ramos, geram novos tableaux, havendo uma relação de dependência entre distintos ramos possuindo a mesma variável livre. Propomos a associação da unificação ao método dos tableaux sem criar dependência entre os seus ramos, e sem necessidade de geração de mais de uma árvore de refutação durante o procedimento de prova, mantendo, desta forma, a característica básica do procedimento, isto é, a geração de uma árvore de refutação a partir de um tableau inicial através da aplicação de regras.

O refinamento altera também o processo de construção do tableau inicial, onde a função de inicialização, além substituir todas as variáveis livres do conjunto inicial de fórmulas por novas constantes e retirar os quantificadores vácuos de cada uma destas fórmulas, também realiza uma renomeação de variáveis, de modo que variáveis sob escopo de quantificadores diferentes possam ser mais facilmente identificadas. Isto é feito para facilitar o processo de unificação.

O processo de unificação modificado trata do ponto crucial desta alteração. No momento da expansão de uma fórmula universal obtemos, através deste processo, uma determinada lista de termos, que nos dirá para quais termos a fórmula universal deve ser instanciada, e se existe a necessidade ou não de repetição desta fórmula no ramo em expansão.

Utilizamos uma série de conceitos relativos ao processo de unificação, as quais podem ser encontradas em [Lloyd 1987]. Basicamente, o algoritmo de unificação padrão varre duas *expressões* que serão unificadas da esquerda para direita, pesquisando a primeira *discordância*, isto é, diferença entre essas duas expressões. Se uma das duas subexpressões localizadas na primeira discordância é uma variável e a outra expressão não contém esta variável, então a *designação* da subexpressão a variável é adicionada a *substituição* que está sendo construída. As duas expressões são instanciadas para a nova substituição e o processo é iterado. Se nenhuma das subexpressões é uma variável, ou se uma subexpressão é uma variável e a outra contém essa variável, então a unificação falha e o algoritmo responde que as expressões não são unificáveis. A unificação obtém sucesso se nenhuma discordância é encontrada, isto é, se as duas expressões originais, através das substituições, tornaram-se idênticas.

Aqui o processo de unificação é executado entre duas fórmulas atômicas assinaladas opostas, onde uma deve ser uma subfórmula positiva e a outra deve ser uma subfórmula negativa. A primeira dessas duas fórmulas atômicas provém de uma listas de fórmulas atômicas assinaladas extraídas da fórmula universal e, a segunda provém de uma lista de fórmulas atômicas assinaladas extraídas do ramo onde a fórmula universal figura. Abaixo definimos o que entendemos por subfórmula positiva, subfórmula negativa, fórmula assinalada, lista de fórmulas atômicas assinaladas de uma fórmulas, e lista de fórmulas atômicas assinaladas de um conjunto de fórmulas.

3.1 Definição: Dizemos quando uma dada primeira fórmula é uma ***subfórmula positiva*** ou uma ***subfórmula negativa*** de uma dada segunda fórmula através das seguintes cláusulas:

- **P** é subfórmula positiva de **P**, $\mathbf{Q} \to \mathbf{P}$, $\mathbf{P} \wedge \mathbf{Q}$, $\mathbf{P} \vee \mathbf{Q}$, $\mathbf{Q} \wedge \mathbf{P}$, $\mathbf{Q} \vee \mathbf{P}$, $\forall \boldsymbol{x}\, \mathbf{P}$ e $\exists \boldsymbol{x}\, \mathbf{P}$;
- **P** é subfórmula negativa de $\neg\mathbf{P}$ e $\mathbf{P} \to \mathbf{Q}$;
- Se **P** é subfórmula positiva de **Q** e **Q** é sufórmula positiva de **R**, então **P** é subfórmula positiva de **R**;
- Se **P** é subfórmula positiva de **Q** e **Q** é sufórmula negativa de **R**, então **P** é subfórmula negativa de **R**;
- Se **P** é subfórmula negativa de **Q** e **Q** é sufórmula positiva de **R**, então **P** é subfórmula negativa de **R**;
- Se **P** é subfórmula negativa de **Q** e **Q** é sufórmula negativa de **R**, então **P** é subfórmula positiva de **R**.

3.13 Definição: ***x*** é uma **v*ariável universal*** em uma fórmula **P** se uma das seguintes condições for satisfeita:

- existe uma subfórmula positiva $\forall \boldsymbol{x}\, \mathbf{Q}$ de **P** tal que x é livre em **Q**;
- existe uma subfórmula negativa $\exists \boldsymbol{x}\, \mathbf{Q}$ de **P** tal que x é livre em **Q**.

Em ambos os casos, todas as ocorrências livres de ***x*** em **Q** são ditas ocorrências universais de ***x*** em **P**.

3.14 Definição: ***x*** é uma **v*ariável existencial*** em uma fórmula **P** se uma das seguintes condições for satisfeita:

- existe uma subfórmula positiva $\exists \boldsymbol{x}\, \mathbf{Q}$ de **P** tal que x é livre em **Q**;
- existe uma subfórmula negativa $\forall \boldsymbol{x}\, \mathbf{Q}$ de **P** tal que x é livre em **Q**.

Em ambos os casos, todas as ocorrências livres de ***x*** em **Q** são ditas ocorrências existenciais de ***x*** em **P**.

3.15 Notação: Usamos a letra **n** seguida de um numeral inteiro positivo como notação para aquilo que denominamos ***constantes existenciais***, as quais usamos no processo de unificação para substituir as variáveis existenciais de uma fórmula.

3.16 Definição: ***Fórmula assinalada*** é um par cujo primeiro componente é um dos sinais "+" ou "-", e o segundo componente é uma fórmula.

3.17 Definição: Obtemos o que chamamos de ***lista de fórmulas atômicas assinaladas de uma fórmula*** **P** através do seguinte processo:

1º) separamos o conjunto **A** de todas as fórmulas atômicas assinaladas (**s**, **Q**) tais que:

- se **s** = "+", então **Q** é subfórmula atômica positiva de **P**;
- se **s** = "-", então **Q** é subfórmula atômica negativa de **P**;

2º) substituímos cada uma das ocorrências existenciais de variáveis em **P** por novas constantes existenciais em todas as fórmulas assinaladas de **A**, onde ocorrências de uma mesma variável existencial são substituídas por uma mesma constante existencial, e ocorrências de diferentes variáveis existenciais são substituídas por diferentes constantes existenciais, obtendo assim a lista desejada.

3.18 Definição: Obtemos o que chamamos de ***lista de fórmulas atômicas assinaladas de uma dada coleção*** Γ ***de fórmulas com respeito a uma primeira lista*** ζ ***de fórmulas assinaladas*** através do seguinte processo:

1º) separamos o conjunto **A** de todas as fórmulas atômicas assinaladas (**s**, **Q**) tais que:

- se **s** = "+", então **Q** é subfórmula atômica positiva de alguma fórmula de Γ;
- se **s** = "-", então **Q** é subfórmula atômica negativa de alguma fórmula de Γ;

2º) substituímos cada uma das ocorrências de variáveis existenciais de alguma fórmula de Γ por novas constantes existenciais, distintas das constantes existenciais que ocorrem em ζ, em todas as fórmulas assinaladas de **A**, onde ocorrências de uma mesma variável existencial são substituídas por uma mesma constante existencial, e ocorrências de diferentes variáveis existenciais são substituídas por diferentes constantes existenciais, obtendo assim a lista desejada.

3.19 Definição: Sejam Γ e Γ' duas coleções de fórmulas atômicas assinaladas. A ***coleção de substituições obtidas de*** Γ e Γ', ***nesta ordem***, é o conjunto de todas as substituições que unificam fórmulas componentes de fórmulas atômicas assinaladas de Γ e Γ' com sinais diferentes.

Assim, para o terceiro refinamento, conservam-se as regras expostas no primeiro refinamento, com exceção da regra para fórmulas universais, substituída pelos passos enumerados a seguir:

1º) obtemos a coleção Γ de fórmulas atômicas assinaladas de $\forall$***x*** **P**, tal que ***x*** é livre nessas fórmulas;

2º) obtemos a coleção Γ' de fórmulas atômicas assinaladas de todas as fórmulas do ramo, com respeito a Γ, dos nós que ainda não foram marcados, e cujas fórmulas ainda não foram comparadas com $\forall$***x*** **P**;

3º) obtemos a coleção de substituições Σ, obtida de Γ e Γ';

4º) caso Σ não possua nenhuma substituição com elementos do tipo ***x***/**t**, então, se $\forall$***x*** **P** não foi instanciado, acrescentamos o tableau da **Figura 4**, caso contrário nada deve ser acrescentado;

5º) caso Σ possua nenhuma substituição com elementos do tipo ***x***/**t**, então reunimos todos os termos $\mathbf{t_1}$,..., $\mathbf{t_n}$, tais que ***x***/$\mathbf{t_i}$ (**i** = 1,..., **n**) figura em algum elemento de Σ, onde os $\mathbf{t_i}$'s são termos fechados sem constantes existenciais;

6º) separamos de $\mathbf{t_1}$,..., $\mathbf{t_n}$ os termos $\mathbf{t_{i1}}$,..., $\mathbf{t_{ik}}$, tais que **P**(***x***/$\mathbf{t_{i1}}$),..., **P**(***x***/$\mathbf{t_{ik}}$) não figuram no ramo, a menos de congruência;

7º) caso exista algum ***x***/**t** figurando em algum elemento de Σ (descrito no passo 3), tal que **t** possui uma constante existencial, então acrescentamos no ramo o tableau da **Figura 2** caso **k** = 0 e $\forall$***x*** **P** ainda não tenha sido instanciado, ou o tableau da **Figura 3** nos demais casos;

8º) caso não exista nenhum ***x***/**t** figurando em algum elemento de Σ, tal que **t** possui uma constante existencial ou uma variável, então acrescentamos o tableau da **Figura 5**;

7º) caso não exista nenhum ***x***/**t** figurando em algum elemento de Σ, tal que **t** possui uma constante existencial, então procuramos por algum ***x***/**t** figurando em algum elemento de Σ tal que **t** possui uma variável ***y***, e por duas fórmulas assinaladas **P** e **Q** de Γ', com sinais distintos, tal que exista um termo **t'** e uma substituição θ em Σ que unificando **P** e **Q**, de modo que ***y***/**t'** pertence a θ, e **t**' possui uma constante existencial ou é um termo que não possui nenhuma instância fechada que já tenha sido instanciada, com respeito a ***y***, pela fórmula do ramo que contém ***y***;

10º) se a busca executada no passo anterior foi bem sucedida, então acrescentamos ao ramo considerado o tableau da Figura 2 caso k = 0 e $\forall$x P ainda não tenha sido instanciado, ou o tableau da

Figura 3, nos demais casos;

11º) se a busca executada no 8º passo não foi bem sucedida, então acrescentamos ao ramo considerado o tableau da **Figura 4** caso **k** = 0 e $\forall$***x*** **P** ainda não tenha sido instanciado, ou o tableau da **Figura 5**, nos demais casos.

Na **Figura 2** e na **Figura 4**, **P**(***x***/$\mathbf{c}_1$)' é uma renomeação completa de **P**(***x***/$\mathbf{c}_1$) para uma coleção de variáveis que não ocorrem no ramo. Na

Figura 3 e na **Figura 5**, **P**(***x***/$\mathbf{t}_{i1}$)',..., **P**(***x***/$\mathbf{t}_{ik}$)' são, respectivamente, renomeações completas de **P**(***x***/$\mathbf{t}_{i1}$),..., **P**(***x***/$\mathbf{t}_{ik}$) para coleções de variáveis disjuntas duas a duas que não ocorrem no ramo.

$$\begin{array}{c} P(x/c_1)' \\ | \\ \forall x\ P \end{array}$$

Figura 2: Tableau inserido pela primeira expansão de uma fórmula $\forall$x **P** quando não ocorrem unificações com termos fechados, e é necessária a repetição da fórmula universal.

$$P(x/t_{i1})'$$
$$\ldots$$
$$P(x/t_{ik})'$$
$$|$$
$$\forall x\ P$$

Figura 3: Tableau inserido pela primeira expansão de uma fórmula ∀x **P** quando ocorrem unificações com termos fechados, ou em qualquer outra expansão, sendo que em qualquer um desses casos é necessária repetição da fórmula universal.

$$P(x/c_1)'$$

Figura 4: Tableau inserido pela primeira expansão de uma fórmula ∀x **P** quando não é necessária a repetição da fórmula universal.

$$P(x/t_{i1})'$$
$$\ldots$$
$$P(x/t_{ik})'$$

Figura 5: Tableau inserido pela primeira expansão de uma fórmula ∀x **P** quando existe pelo menos uma unificação com termos fechados, e não é necessária a repetição da fórmula universal.

4 – Exemplificando o Sistema de Tableaux para Lógica Clássica e o Sistema de Tableaux para Lógica Clássica Usando o Método da Unificação

Para evidenciar as diferenças entre o método dos tableaux tradicional e o método dos tableaux com unificação apresentaremos um esquema do comportamento desses dois sistemas para o mesmo problema.

Problema: $Pc_1 \wedge Qc_2, \forall x \neg Sx \vdash \exists x(Px \vee Qx)$

Desenvolvimento utilizando o Método dos Tableaux Tradicional

√1) $Pc_1 \wedge Qc_2$

√2) $\forall x \neg Sx$

√3) $\neg \exists x(Px \vee Rx)$

√4) Pc_1

√5) Qc_2

√6) Sc_1

√7) Sc_2

√8) $\forall x \neg Sx$

√9) $\forall x \neg (Px \vee Rx)$

√10) $\forall x \neg Sx$

√11) $\neg (Pc_1 \vee Rc_1)$

12) $\neg (Pc_2 \vee Rc_2)$

13) $\forall x \neg (Px \vee Rx)$

14) $\forall x \neg Sx$

15) $\neg Pc_1$

X

Desenvolvimento utilizando o Método dos Tableaux com Unificação

√1) $Pc_1 \wedge Qc_2$

√2) $\forall x \neg Sx$

√3) $\neg \exists x(Px \vee Rx)$

√4) Pc_1

√5) Qc_2

√6) $\forall x \neg (Px \vee Rx)$

√7) $\neg (Pc_1 \vee Rc_1)$

8) $\neg Pc_1$

X

Em ambos os casos, os nós 1, 2 e 3 representam a árvore inicial para o desenvolvimento do tableaux e cada nó marcado com um √, significa que este nó já foi processado, isto é, expandido se isto for possível.

No método tradicional o algoritmo, ao processar uma fórmula universal, simplesmente a instancia para todas as constantes do ramo onde a mesma figura, repete esta fórmula, e prossegue com a expansão do próximo nó.

No método com unificação, ao processar uma fórmula universal o algoritmo realiza o processo de unificação entre as subfórmulas atômicas da fórmula universal e as subfórmulas atômicas das demais fórmulas do ramos, para encontrar as constantes que são mais apropriadas ao fechamento do ramo. No exemplo, quando o algoritmo processa o nó 2, verifica que não existe unificação entre Sx e as demais subfórmulas atômicas do ramo, então não é necessário a instanciação nem a repetição da fórmula deste nó. Já no processamento do nó 6, o algoritmo encontra uma unificação entre a subfórmula atômica positiva Pc_1 do nó 1, e a subfórmula atômica negativa Px do nó 6, logo, x deve ser substituído por c_1. A fórmula do nó 6 não é repetida por não existirem unificações entre as subfórmulas atômicas deste nó com outras subfórmulas atômicas que contém constantes existenciais ou variáveis universais, que ainda não foram instanciadas, ou que por sua vez unificam com outras constantes existenciais.

5 – REFERÊNCIAS

[Bittencourt 1997]
Bittencourt, Guilherme. *Inteligência Artificial: Ferramentas e Teorias*. Ed. da UFSC, Florianópolis, 1998.

[Bell & Machover 1993]
Bell, John Lane & Machover, Moshé. *A Course in Mathematical Logic*. Elsevier Science Publishers B.V., Amsterdam, 1993.

[Buchsbaum 1988]
Buchsbaum, Arthur. *Um Método Automático de Prova para a Lógica Paraconsistente*. Dissertação de Mestrado do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1988.

[Buchsbaum 1995]
Buchsbaum, Arthur. *Lógicas da Inconsistência e da Incompletude: Semântica e Axiomática*. Tese de Doutorado, Pontificia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1988.

[Buchsbaum & Pequeno 1990]
Buchsbaum, Arthur & Pequeno Tarcisio. *O Método dos Tableaux Generalizado e sua Aplicação ao Raciocínio Automático em Lógicas Não Clássicas*. O que nos faz pensar – Cadernos do Departamento de Filosofia da Pontifícia Universidade Católica do Rio de Janeiro, nº 3, setembro de 1990.

[Buchsbaum & Pequeno 1991]
Buchsbaum, Arthur & Pequeno Tarcisio. Uma família de Lógicas Paraconsistente e/ou Partacompletas com Semânticas Recursivas. Monografias em Ciência da Computação nº 5/91. Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, 1991. Republicado por Coleção Documentos Série de Lógica e Teoria da Ciência nº 14. Instituto de Estudos Avançados, Universidade de São Paulo, 1993.

[Buchsbaum & Pequeno 1994]
Buchsbaum, Arthur & Pequeno Tarcisio. *Automated Deduction with Non Classical Negations*. Proceedings of 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods - 1994 - pp. 51-64.

[Chang & Lee 1973]
Chang, Chin-Liang & Lee, Richard Char-Tung. *Simbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.

[Church 1936]
Church, Alonzo. *A Note on the Entscheidungsproblem*. The Journal of Symblic Logic, vol.1, pp 40-41, 101-102, 1936.

[Enderton 1972]
Enderton, Herbert B.. *A Mathematical Introduction to Logic*, Academic Press, 1972.

[Fiitting 1990]
Fitting, Melvin. *First-Order Logic and Automated Theorem Proving.* Springer Verlag, 1990.

[Lloyd 1987]
Lloyd, J. W.. *Foundations of Logic Programming*. Springer Verlag, 1987.

[Loveland 1978]
Loveland, Donald W.. *Automated Theorem Proving*. North-Holland Publishing Company, 1978.

[Reeves 1983]
Reeves, S. V.. *An Introduction to Semantic Tableaux. Department of Computer Science*. CMS-55, University of Essex, 1983.